國立交通大學

資訊工程學系

博 士 論 文

具隱私性之資料擷取：忘卻式傳輸與播送

Privacy-Preserving Data Retrieval: Oblivious Transfer and Cast

研 究 生：朱成康

指導教授：曾文貴　教授

中 華 民 國 九 十 七 年 六 月

具隱私性之資料擷取：忘卻式傳輸與播送

Privacy-Preserving Data Retrieval: Oblivious Transfer and Cast

研 究 生：朱成康　　Student：Cheng-Kang Chu

指導教授：曾文貴　博士　　Advisor：Dr. Wen-Guey Tzeng

國 立 交 通 大 學

資 訊 工 程 學 系

博 士 論 文

A Dissertation

Submitted to Department of Computer Science

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

具隱私性之資料擷取：忘卻式傳輸與播送

學生：朱成康　　指導教授：曾文貴 博士

國立交通大學資訊工程學系

摘要

具隱私性之資料擷取一直是密碼學上一個重要的課題，它可以讓使用者在取得資料的同時，也保有隱私性。在本論文裡面，我們對於這個課題有兩個貢獻。首先，我們提出了一些有效率的忘卻式傳輸機制。忘卻式傳輸包含了一個傳送者與一個接收者，傳送者有一些訊息在手上，而接收者想取得其中的某幾個訊息。對此，安全性的要求在於接收者只能取得他想要的那部分訊息，而且不能讓傳送者知道他的選擇。忘卻式傳輸的發展分為一些類別，例如 2 選 1、n 選 1、n 選 k、動態、條件忘卻式傳輸等等。本論文裡我們提出數個 n 選 k 及一個動態忘卻式傳輸機制。

接著，我們提出一個新的概念，稱為『條件忘卻式播送』。與忘卻式傳輸不同，這種系統有三個參與者：一個傳送者及兩個接收者。傳送者有一個訊息，而接收者擁有他們各自的秘密值。當接收者的兩

個秘密值符合某個我們定義的條件，那麼兩個接收者即可從傳送者那邊取得訊息。此概念安全性的要求在於，每位參與者都不能知道另外兩位手中的秘密。同時，我們也提出一些條件忘卻式播送的變形。對於這些概念及變形，我們提出了一些實際的機制，針對的是「相等」、「不相等」、「大於」等基本的條件，有了這些基本的機制，便可延伸設計出更複雜的密碼協定。

關鍵字：忘卻式傳輸，動態忘卻式傳輸，忘卻式播送，條件忘卻式播送

Privacy-Preserving Data Retrieval: Oblivious Transfer and Cast

Student: Cheng-Kang Chu Advisor: Dr. Wen-Guey Tzeng

Department of Computer Science

National Chiao Tung University

Abstract

Privacy-preserving data retrieval is an important cryptographic issue. It allows users to privately obtain some data they need. In this thesis, we have a two-fold contribution to privacy-preserving data retrieval. Firstly, we give some efficient oblivious transfer (OT) constructions. OT involves two parties: a sender S and a receiver R. The sender S has several messages of which the receiver R wants to get some. The security requirement is that S wants R to get the messages of his choices only, and R does not want S to know what he chooses. OT was developed in several types, such as 1-out-of-2 OT, 1-out-of-n OT, k-out-of-n OT, adaptive OT, conditional OT. We propose several efficient k-out-of-n OT schemes and an adaptive OT scheme.

Then, we introduce a new notion of *conditional oblivious cast* (COC), which involves three parties: a sender S and two receivers A and B. Receivers A and B own secrets x and y, respectively, and the sender S holds a message m. In a COC scheme for a predicate Q (Q-COC), A and B get m from S if and only if $Q(x, y) = 1$. The security requirement is that each

party does not leak his secret to the other two parties except the information that can be inferred from the result. We also extend COC to different settings: only one receiver gets the message or S sends two messages at a time. We give definitions for COC and propose COC schemes for "equality", "inequality", and "greater than" predicates. These are fundamental schemes for constructing more complex secure multi-party protocols. Finally, we provide new constructions of three-party *oblivious cast*, which are more efficient in communication complexity than previous schemes.

Keywords: oblivious transfer, adaptive oblivious transfer, oblivious cast, conditional oblivious cast

誌　　謝

首先感謝我的指導老師曾文貴教授，在我碩士班及博士班總共九年的學習過程中，帶領我深入密碼學的領域，老師認真積極的教學態度，使我受益良多。另外，我要感謝口試委員，成大賴溪松教授、台科大吳宗成教授、清大孫宏民教授、警大王旭正教授、交大陳榮傑教授、交大謝續平教授、交大蔡錫鈞教授與交大黃世昆教授，在論文上給我許多建議與指導，讓我的論文更加完善。除此之外，我也要感謝學長胡智明教授特別北上來給我鼓勵，以及實驗室學弟妹，孝盈、煥宗和其他的碩士班同學的幫忙。

最後，我要感謝我的家人及季穎，不論在精神或物質上都給我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，謹以此文獻給所有我想要感謝的人。

Contents

Chapter 1

Introduction

Electronic commerce (eCommerce) is a very popular and important service on Internet nowadays. People can buy anything they need by just clicking buttons at home. In most situations, users' personal data (e.g. credit card numbers) are well encrypted. However, there is no guarantee of users' privacy. Why should we tell these merchants which music we like or which book we want to buy? Could we conceal our choices from the merchant?

Privacy-preserving data retrieval allows users privately obtaining some data they need from a sender. In many applications, users' privacy should be well protected. For example, researchers should be able to query a database without revealing which topics they are interested in, and bidders should be able to bid for a secret without exposing their prices.

In fact, we can use a general secure multi-party computation protocol to fulfill the above requirements. However, we are seeking for more efficient solutions of some specific cases. In this thesis, we discuss privacy-preserving data retrieval for two-party and three-party cases. We focus on oblivious

transfer (OT) and conditional oblivious cast (COC) protocols for these two cases, respectively. In this chapter, we first introduce the concepts of OT and COC, and then state our results to the both notions.

1.1 Oblivious Transfer

Oblivious transfer (OT) is an important cryptographic protocol. It protects users' privacy from exposing their choices and preserves merchants' security. In other words, by invoking OT, a buyer only gets the paid goods, and a merchant doesn't know which ones are chosen. Moreover, OT is an important building block of cryptographic protocols [GV87, Kil88, Yao86].

An oblivious transfer scheme involves two parties: a sender S and a receiver R. S has several messages and R wants to obtain some of them via interaction with S. There are two security requirements for oblivious transfer: (1) R obtains the chosen messages only; (2) S doesn't know which messages are chosen. The original OT was proposed by Rabin [Rab81]. Then it was developed in the following main types:

- Rabin's OT: S sends a message to R, and R gets the message with probability 1/2. On the other hand, S doesn't know whether R gets the message or not.

- 1-out-of-2 OT (OT_2^1): S has two messages m_1 and m_2, and allows R to obtain exactly one of them. In addition, S remains oblivious to R's choice.

- 1-out-of-n OT (OT_n^1): An extension of OT_2^1 for the case that S has n messages.

- k-out-of-n OT (OT_n^k): Similar to OT_n^1 except that R obtains k out of n messages from S.

We are concerned about the most general case - OT_n^k in this work. A straightforward solution for OT_n^k is to run OT_n^1 k times independently. However, this needs k times the cost of OT_n^1. So we hope to develop a direct construction for OT_n^k. Security of OT is also an interesting issue. Since it is impossible to provide unconditional security for both sender and receiver, we consider the unconditional security for each party separately, and the security of the other side is computational. Users can choose appropriate schemes in various applications with different security settings.

Oblivious transfer with adaptive queries (Adpt-OT) allows R to query messages one by one adaptively [NP99b, OK04]. For this setting, S first commits messages to R in the commitment phase. Then in the transfer phase, R makes queries of messages one by one. Adaptive OT is natural and has many applications, such as oblivious search, oblivious database queries, private information retrieval.

1.2 Conditional Oblivious Cast

Instead of depending on R's choices solely, conditional oblivious transfer (COT) [COR99, BK04] lets R get the message m under some condition.

For example, in a Q-COT scheme, S and R have their own secrets x and y respectively, and R gets m from S if and only if $Q(x, y) = 1$. Furthermore, S can not get the result of the condition on x and y.

Three-party oblivious cast (OC) [Bla96, FGMO01] is a generalization of OT to the three-party case: one sender S with a secret bit and two receivers A and B. The bit is received by exactly one receiver, each with probability 1/2. We further generalize oblivious cast to *conditional oblivious cast* (COC), where the receivers A and B have their own secrets and can get messages from S if and only if some condition of their secrets holds. The main idea of COC is to separate the role of the secret holder from the message holder S in COT. A COC scheme that meets the requirements of our definitions can be easily transferred to a COT scheme.

For some predicate Q, let the receivers A and B hold secrets x and y, respectively. We discuss three types of COC:

- Q-COC-I: S has a message m. A and B get m if and only if $Q(x, y) = 1$;
- Q-COC-II: S has two messages m_0 and m_1. A and B get m_1 if and only if $Q(x, y) = 1$ and m_0 if and only if $Q(x, y) = 0$;
- Q-COC-III: S has a message m. A gets m if and only if $Q(x, y) = 1$ and B gets m if and only if $Q(x, y) = 0$.

In all types of COC, the secret x (resp. y) can not be revealed to either the receiver B (resp. A) or the sender S.

COC not only covers all functionalities of COT, but also broadens the range of its applications. We provide three examples here:

- *Three-party priced oblivious transfer*: Aiello et al. [AIR01] introduced the notion of "priced oblivious transfer", which protects the privacy of a customer's purchase from a vendor. In their setting, the buyer needs to deposit some amount of money in each vendor. This is not practical if a buyer wants to purchase various goods from various vendors. By our COC schemes, we have a generalized priced OT such that the buyer can deposit the money in one bank only. When the buyer wants to buy an item from a vendor, he sends the corresponding price and the bank sends the buyer's current balance in encrypted form to the vendor. The vendor then sends the item so that the buyer can get it if the balance is above the price.

- *Oblivious two-bidder system*: A party S has a secret for selling, and A and B are two bidders. The winner can obtain the secret from S directly. At the end, S has no idea who the winner is. This system can be constructed from COC-III for the "greater than" predicate immediately.

- *Oblivious authenticated information retrieval*: A user can anonymously get messages from a server if he passes the authentication procedure provided by another party. For instance, consider a mobile downloading service which lets users download music, games, movies, etc. from

various merchants without revealing their identities. Assume that a mobile phone has no extra memory to store the subscription information. Users first apply the service to the mobile communication company, and the company provides an encrypted subscription list of IMSIs (International Mobile Subscriber Identity) to the content provider. When a user wants to download a game, his mobile phone sends its encrypted IMSI to the content provider. The content provider then sends the game and the user can get it if the IMSI is in the subscription list. In this case, the user's identity (IMSI) is anonymous to the content provider.

1.3 Our Results

In this thesis, we propose several efficient k-out-of-n OT and COC constructions. For OT_n^k, we provide solutions for three different classes: OT with unconditional security of receiver (OT_n^k-I), OT with unconditional security of sender (OT_n^k-II), and OT with adaptive queries (Adpt-OT).

- In the case of unconditional receiver's security, we propose two schemes Semi-OT_n^k-I and Mal-OT_n^k-I with two rounds only, in which R sends $O(k)$ messages to S, and S sends $O(n)$ messages back to R. Semi-OT_n^k-I is secure against semi-honest receivers if the Decisional Diffie-Hellman (DDH) problem holds. Mal-OT_n^k-I can be proved secure against malicious receivers under the Chosen-Target Computational Diffie-Hellman

	Semi-OT$_n^k$-I	Mal-OT$_n^k$-I	[MZV02]	[WZW03]
rounds	2	2	2	3
messages† ($R \to S$)	k	k	$2n$	k
messages† ($S \to R$)	$2n$	$n+k$	$2n$	$n+k$
made to adaptiveness	No	Yes	No	Yes
security proof	Yes	Yes (RO)	No	No

† The number of group elements.

Table 1.1: Comparison between OT$_n^k$ schemes with unconditional receiver's security in communication cost.

problem in the random oracle model. When $k = 1$, the schemes are as efficient as the OT$_n^1$ schemes in [Tze04]. The schemes also have the nice property of universal parameters, that is, each pair of R and S need not hold any secret before performing the protocol. The system parameters can be used by all senders and receivers without any trapdoor specification. Our OT$_n^k$ schemes in this class are the most efficient ones in terms of the communication cost, either in rounds or the number of messages. We summarize the comparison between ours and other efficient works in this class in Table 1.1. Preliminary versions of these results have been previously published [CT05, CT08].

- In the case of unconditional sender's security, we first propose a generic construction Gen-OT$_n^k$-II where any multiplicatively homomorphic encryption scheme whose plaintext space has a prime order can be applied. The query phase of the construction is still two-round. The receiver's security is based on the semantic security of the underlying

encryption scheme only. Then we provide a concrete construction Con-OT$_n^k$-II with only two rounds. The receiver's security is based on the DDH problem and can be proved in the standard model. For some cases of k and n, our Con-OT$_n^k$-II scheme is the most efficient scheme in terms of the communication cost[1]. Preliminary versions of these results have been previously published [CT08].

- For OT with adaptive queries, we extend our Mal-OT$_n^k$-I scheme to an adaptive OT scheme, named Adpt-OT$_n^k$. In this scheme, S first sends $O(n)$ messages to R in one round in the commitment phase. For each query of R, only $O(1)$ messages are exchanged and $O(1)$ operations are performed. Our construction is as efficient as the work of Ogata and Kurosawa [OK04], which is the most efficient adaptive OT scheme up to now. Preliminary versions of these results have been previously published [CT05, CT08].

For COC, we propose COC-I and COC-II schemes for "equality", "inequality", and "greater than" predicates. These are fundamental schemes for constructing more complex secure interactive protocols. Then we also provide a general transformation from COC-II schemes to COC-III schemes. Our schemes are efficiently constructed via homomorphic encryption schemes. We first prove our schemes secure against semi-honest and non-collusive parties, and then modify them to be secure against malicious parties. We also pro-

[1]For the scheme with unconditional sender's security, one may perform the schemes of Lipmaa [Lip05] k times independently to get better efficiency in some cases of k and n.

pose some OC schemes based on our COC-III scheme and homomorphic cryptosystems. Our schemes are more efficient in communication complexity than the previous one. Finally, we give some extensions of COC. Preliminary versions of these results have been previously published [CT06].

Chapter 2

Previous Works

In this chapter we provide a detailed survey of various OT and OC, respectively.

2.1 Oblivious Transfer

In 1981, Rabin [Rab81] first introduced the notion of OT. Then, OT was developed in several basic types:

- Rabin's OT: S sends a message to R, and R gets the message with probability 1/2. On the other hand, S does not know whether R gets the message. Rabin [Rab81] presented an implementation to obliviously transfer one-bit message, based on quadratic roots modulo a composite. Even, Goldreich and Lempel [EGL82] provided another implementation based on general public-key encryptions. Berger, Peralta and Tedrick [BPT84] also proposed a provably secure scheme. Beaver [Bea92] provided another scheme by fixing Boer's [Boe90] insecure implementation. Fischer, Micali and Rackoff [FMR96] also fixed

Rabin's implementation so that it can be proved secure assuming only the factoring problem holds.

- 1-out-of-2 OT (OT_2^1): S has two messages m_1 and m_2, and wishes R to obtain exactly one of them. In addition, S remains oblivious to R's choice. The notion of OT_2^1 originated with the work of Wiesner [Wie83], which is also the beginning of quantum cryptography. Even, Goldreich and Lempel [EGL85] proposed such extension of OT_2^1, in which R gets either message with probability 1/2. Brassard, Crépeau and Robert [BCR86b] give us the new notion of OT_2^1 such that R can get the message at his choosing. Crépeau and Kilian [CK88] provided another bit-OT_2^1 (the messages are only one-bit) scheme from the weakened security assumptions. Then many other OT_2^1 schemes are continually proposed [BM89, CS91, Boe90, SP90, SCP95, BC97, Cré97, Cac98, CMO00, AIR01, Hai04, Kal05, WW06, CS06, Wul07].

- 1-out-of-n OT (OT_n^1): OT_n^1 is an extension of OT_2^1 for the case that S has n messages. Brassard, Crépeau and Robert [BCR86a] first proposed OT_n^1 in the name "all-or-nothing disclosure of secrets" (ANDOS). After that, OT_n^1 has become an important research topic in cryptographic protocol design. Some OT_n^1 schemes are built by invoking basis OT_2^1 several times [BCR86b, BCS96, NP99a], and the others are constructed directly from basic cryptographic techniques [SS90, NR94, Ste98, NP01, Tze04, Lip07, LL07]. Some OT_n^1 schemes derived from

computational private information retrieval (CPIR) have polylogarithmic communication cost [Cha04, Lip05].

- k-out-of-n OT (OT_n^k): R obtains k out of n messages from S. Bellare and Micali [BM89] first proposed an OT_n^{n-1} scheme. Santis, Crescenzo, and Persiano [SCP95] extended their OT_2^1 scheme to an OT_n^k scheme. Ishai and Kushilevitz [IK97] also provided a bit-OT_n^k scheme by invoking OT_2^1 several times. Naor and Pinkas [NP99a] proposed a non-trivial OT_n^k scheme. The scheme invokes a basis OT_2^1 scheme $O(wk \log n)$ times, where $w > \log \delta / \log(k^4/\sqrt{n})$ and δ is the probability that R can obtain more than k messages. It works only for $k \leq n^{1/4}$. Mu, Zhang, and Varadharajan [MZV02] presented some efficient OT_n^k schemes. These schemes are designed from cryptographic functions directly. The most efficient one is a non-interactive one. To be compared fairly, the setup phase of establishing shared key pairs of a public-key cryptosystem should be included. Thus, the scheme is two-round and R and S send each other $O(n)$ messages. However, the choices of R cannot be made adaptive since R's choices are sent to S first and the message commitments are dependent on the choices. Wu, Zhang, and Wang [WZW03] also provided a three-round OT_n^k based on the two-lock cryptosystem. Recently, Green and Hohenberger [GH07] presented an OT_n^k and an adaptive OT schemes in the full-simulation model which needs several additional rounds.

Relations between basic types of OT Crépeau [Cré87] proved that Rabin's OT and OT_2^1 are computationally equivalent. However, it needs to reduce bit-OT_2^1 to string-OT_2^1 first, and then reduce string-OT_2^1 to Rabin's OT. Kurosawa and Koshiba [KK07] provided a direct reduction of string-OT_2^1 to Rabin's OT. Certainly, OT_n^k implies OT_n^1 and OT_n^1 implies OT_2^1 by setting $k = 1$ and $n = 2$, respectively. There are also some reductions from OT_n^1 to OT_2^1 [BCR86b, BCS96, NP99a]. Moreover, OT_n^k can be easily constructed by invoking OT_n^1 k times independently. Therefore, these four basic types of OT are computationally equivalent. Other reductions between OTs are discussed in [CK88, BC97, Cac98, DM99, DKS99, DS01, IKNP03, DFMS04, Nie07].

Besides, there are various types of OT developed in different models and applications:

- Adaptive OT (Adpt-OT): The OT scheme allows R to query messages one by one adaptively, i.e. the ith chosen message depends on the previous $i - 1$ messages. The notion was first introduced by Naor and Pinkas [NP99b]. In one of their schemes (the two-dimensional one), each query needs invoke the basis $\text{OT}_{\sqrt{n}}^1$ scheme twice, in which each invocation of $\text{OT}_{\sqrt{n}}^1$ needs $O(\sqrt{n})$ initialization work. In another scheme, each adaptive query of messages need invoke the basis OT_1^2 protocol $\log n$ times. Ogata and Kurosawa [OK04] proposed an efficient adaptive OT scheme based on the RSA cryptosystem. Each S needs a trapdoor (the RSA modulus) specific to him. The scheme is as efficient

as our Adpt-OT$_n^k$ scheme. But, if the adaptive OT scheme is converted to a non-adaptive one, it needs 3 rounds (In the first round, S sends the modulus N to R). There are some recent works [GH07, CNs07] introducing adaptive OT in the full-simulation model. These schemes achieve the stronger security requirement but loss a degree of efficiency.

- Conditional OT (COT): S holds a secret x and a message m, and R holds a secret y such that R gets m from S if and only if $Q(x, y)$ is evaluated as true for some condition Q. COT was first proposed by Crescenzo, Ostrovsky and Rajagopalan [COR99]. They focus on providing "all-or-nothing" transfer of the message from S to R by the condition. Blake et al. [BK04] strengthened COT to strong COT (SCOT), which provides "1-out-of-2" message transfer from S to R by the condition. Our notion of COC is to split the role of S who owns a message and a secret into two roles of a message sender and a secret holder. The main difference in design techniques is that, in COT, the secure computation is done by S with a masked input and a plain input, whereas the secure computation in our COC schemes is done by S with two masked inputs. A COC scheme that meets the requirements of our definitions can be easily transferred to a COT or SCOT scheme.

- Verifiable OT (VOT): R can verify that the messages he gets from S are indeed the messages S had committed. Crépeau [Cré89] first proposed a basic bit-VOT$_2^1$ scheme. Then the VOT$_n^1$ scheme can be immedi-

ately obtained by the general transformation [BCR86b]. Crépeau, van de Graaf, and Tapp [CvdGT95] provided a more efficient scheme in the name "committed oblivious transfer" by using OT_2^1 and BC (bit commitment) as black boxes. Cramer and Damgård [CD97] presented another solution based on their zero-knowledge proof system. Cachin and Camenisch [CC00] proposed a simple and efficient VOT_2^1 based on the DDH assumption. Ambainis, Jakobsson and Lipmaa [AJL04] provided a VOT_n^1 based on the OT_n^1 scheme proposed in [NP01]. There are also other variants of OT with verifiable property [JS02, Lip03, GMY04, KSV07].

- Distributed OT (DOT): S is divided between several servers, and R must contact a threshold of these servers to perform the OT protocol. Compared to the single-server based OT, DOT has the advantage of better efficiency and security. Gertner and Malkin [GM97] first introduced the notion of DOT and presented an implementation of DOT_n^1. Naor and Pinkas [NP00] provided a solution of DOT_2^1 (it can also be extended to DOT_n^1 by their reduction [NP99a]). Blundo, D'Arco, Santis and Stinson [BDSS02] also constructed some DOT_n^1 schemes. Nikov, Nikova, Preneel and Vandewalle [NNPV02] further generalized the result of Blundo et. al. Tzeng [Tze04] proposed a DOT_n^1 scheme based on his OT_n^1 scheme in the name "threshold oblivious transfer".

- Proxy OT (POT): In addition to S and R, a POT scheme involves a

third party, the proxy, which serves as the receiver's proxy for learning the output. Naor, Pinkas, and Sumner [NPS99] first introduced this notion and provided an implementation of POT_2^1. Juels and Szydlo [JS02] also proposed another verifiable POT scheme. These two POT schemes are both used in the auction systems.

- OT in the bounded storage model: The bounded storage model, first introduced by Maurer [Mau90], assumes that there is only a bound on the adversary's *storage* capacity, i.e. no limitation on the computational power of the adversary. In the bounded storage model, provably everlasting security can be achieved efficiently without any complexity assumption [CM97, AR99, ADR02, DR02, DM02, Lu02, Vad03]. The first OT_2^1 scheme in this model was proposed by Cachin, Crépeau and Marcil [CCM98]. They put the storage bound on R (the case where the bound is placed on S is equivalent by the reversibility of OT [CS91]). The storage requirement for the two parties is $O(N^{2/3})$, where N is the (very large) number of public random bits, and the successful probability of dishonest R is at most $O(N^{-1/3})$, which is not small enough. Ding [Din01] provided a more efficient OT_2^1 scheme with $O(\sqrt{kN})$ storage requirement and $2^{-O(k)}$ successful probability of dishonest R who stores $O(N)$ bits, where k is a security parameter. Hong, Chang and Ryu [HCR02] extended Ding's work to get an OT_n^1 scheme in the bounded storage model. However, all these schemes require a lot

of interaction. Ding, Harnik, Rosen and Shaltiel [DHRS04] provided a constant round OT_2^1 scheme. The two parties in their work need only exchange 5 messages. They also improved other parameters, such as the number of bits transferred and the probability of immaturely aborting the protocol due to the failure.

There are also some related works designed for particular considerations [Bea95, Bea96, GM00, HKN+05, LL06]. For the relation between OT and other cryptographic primitives, Kilian [Kil88] showed that OT is complete in the sense that every secure multiparty computation can be implemented using OT. Impagliazzo and Luby [IL89] proved that OT implies the existence of one-way functions (OWF). Moreover, Impagliazzo and Rudich [IR89] showed that OT is probably stronger than OWF, that is, the construction of OT using OWF as a black box is as hard as separating P from NP. Gertner, Kannan, Malkin, Reingold and Viswanathan [GKM+00] studied the relationships among public-key encryption (PKE) and OT. They showed that PKE and OT are incomparable under black-box reductions.

2.2 Oblivious Cast

Blaze [Bla96] introduced the notion of k-out-of-n OC, e.g. there are n receivers and k of them get the message from the sender. He provided a concrete construction by using the blind signature scheme and anonymous communication mechanism. Fitzi et al. [FGMO01] treated OC as a primitive

for the secure multi-party computation. They proved that three-party OC is complete when the corrupted parties is less than one half.

Chapter 3

k-out-of-n Oblivious Transfer

We propose several OT_n^k schemes in different security classes in this chapter.

3.1 Preliminaries

Involved parties. An OT scheme has two involved parties: a sender and a receiver. Both are polynomial-time-bounded probabilistic Turing machines (PPTM). A party is semi-honest (or passive, honest-but-curious) if it does not deviate from the steps defined in the protocol, but tries to compute extra information from received messages. A party is malicious (or active) if it can deviate from the specified steps in any way in order to get extra information.

A malicious sender may cheat in order or content of his possessed messages. To prevent the cheat, we can require the sender to commit the messages first [CNs07, GH07]. But we don't stress on this issue here because we should handle it at the application level above the OT. Actually, most merchants which play the roles of senders act honestly in order to keep their reputation. Therefore, we consider the OT scheme with semi-honest senders

and semi-honest/malicious receivers only.

Indistinguishability. Two distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are computationally indistinguishable if for any PPTM D and any polynomial $p(n)$, there is n_0 such that for every $n \geq n_0$

$$|\Pr_{x\leftarrow X_n}[D(x)=1] - \Pr_{x\leftarrow Y_n}[D(x)=1]| \leq 1/p(n).$$

Ideal Model. In the ideal model, the sender sends all messages and the receiver sends all choices to a trusted third party (TTP). The TTP then sends the chosen messages to the receiver. This is the securest way to implement OT.

Security model. Assume that S holds n messages $m_1, m_2, \ldots, m_n$ and R has k choices $\sigma_1, \sigma_2, \ldots, \sigma_k$. For OT with honest sender and honest receiver, we have the following security requirements:

1. Receiver's security - indistinguishability: for any two different sets of choices $C = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ and $C' = \{\sigma'_1, \sigma'_2, \ldots, \sigma'_k\}$, the transcripts, corresponding to C and C', received by the sender are indistinguishable. If the received messages of S for C and C' are identically distributed, the choices of R are unconditionally secure.

2. Sender's security - indistinguishability: for any set of choices $C = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, the unchosen messages should be indistinguishable from the random ones. If the ciphertexts of unchosen messages are uniformly distributed for R, the security of S is unconditional.

On the other hand, the OT with honest sender and malicious receiver should meet the following security requirements:

1. Receiver's security - indistinguishability: the same as the case of the semi-honest receiver.

2. Sender's security - compared with the ideal model: for any receiver R in the real OT scheme, there is a simulator R' in the ideal model such that the outputs of R and R' are indistinguishable.

Computational model. Let $\mathbb{G}_q$ be a subgroup of $\mathbb{Z}_p^*$ with prime order q, where $p = 2q + 1$ is also prime. Let g be a generator of $\mathbb{G}_q$. We use g^x to denote the abbreviation of $g^x \bmod p$, where $x \in \mathbb{Z}_q$. Let $x \in_R X$ denote that x is chosen uniformly and independently from the set X.

Computational assumptions. We assume that the hardness of the Decisional Diffie-Hellman (DDH) problem and the Chosen-Target Computational Diffie-Hellman (CT-CDH) problem.

Assumption 1 (Decisional Diffie-Hellman (DDH)) *The following two distribution ensembles are computationally indistinguishable:*

- $Y_1 = \{(g, g^a, g^b, g^{ab})\}_{\mathbb{G}_q}$, *where* g *is a generator of* $\mathbb{G}_q$, *and* $a, b \in_R \mathbb{Z}_q$.
- $Y_2 = \{(g, g^a, g^b, g^c)\}_{\mathbb{G}_q}$, *where* g *is a generator of* $\mathbb{G}_q$, *and* $a, b, c \in_R \mathbb{Z}_q$.

The CT-CDH assumption, introduced by Boldyreva [Bol03], is analogous to the chosen-target RSA inversion assumption defined by Bellare, et al. [BNPS01]

S, R	a sender and a receiver
$\mathbb{G}_q$	a cyclic group with prime order q
g, h	generators of group $\mathbb{G}_q$
m_i's	sender's messages
σ_i's	receiver's choices
f_1, f_2	polynomials
a_i's	coefficients of f_1
b_i's	coefficients of f_2
H_1, H_2	hash functions
$\mathcal{E}$	an encryption scheme
G, E, D	key generation, encryption and decryption functions
pk, sk	a pair of public key and secret key
$\mathcal{M}$	message space of an encryption scheme
λ	a security parameter

Table 3.1: Common notations for our OT constructions

Assumption 2 (Chosen-Target Computational Diffie-Hellman (CT-CDH)) *Let $H_1 : \{0,1\}^* \to \mathbb{G}_q$ be a cryptographic hash function. The adversary A is given input (q, g, g^x, H_1) and two oracles: target oracle $T_G(\cdot)$ that returns a random element $w_i \in \mathbb{G}_q$ at the ith query and helper oracle $H_G(\cdot)$ that returns $(\cdot)^x$. Let q_T and q_H be the number of queries A made to the target oracle and helper oracle, respectively. The probability that A outputs k pairs $((v_1, j_1), (v_2, j_2), \ldots, (v_k, j_k))$, where $v_i = (w_{j_i})^x$ for $i \in \{1, 2, \ldots, k\}$, $q_H < k \leq q_T$, is negligible.*

Finally, we list some common notations used in this chapter in Table 3.1 for quick reference.

- System parameters: $(g, h, \mathbb{G}_q)$;
- S has messages: $m_1, m_2, \ldots, m_n \in \mathbb{G}_q$;
- R's choices: $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$;

1. R chooses two polynomials $f_1(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} + x^k$ and $f_2(x) = b_0 + b_1 x + \cdots + b_{k-1}x^{k-1} + x^k$ where $a_0, a_1, \ldots, a_{k-1} \in_R \mathbb{Z}_q$ and $b_0 + b_1 x + \cdots + b_{k-1}x^{k-1} + x^k \equiv (x - \sigma_1)(x - \sigma_2)\cdots(x - \sigma_k) \bmod q$.
2. $R \longrightarrow S : A_0 = g^{a_0}h^{b_0}, A_1 = g^{a_1}h^{b_1}, \ldots, A_{k-1} = g^{a_{k-1}}h^{b_{k-1}}$.
3. S computes $c_i = (g^{r_i}, m_i B_i^{r_i})$ where $r_i \in_R \mathbb{Z}_q$ and $B_i = g^{f_1(i)}h^{f_2(i)} = A_0 A_1^i \cdots A_{k-1}^{i^{k-1}}(gh)^{i^k} \bmod p$, for $i = 1, 2, \ldots, n$.
4. $S \longrightarrow R$: $c_1, c_2, \ldots, c_n$.
5. Let $c_i = (U_i, V_i)$. R computes $m_{\sigma_i} = V_{\sigma_i}/U_{\sigma_i}^{f_1(\sigma_i)} \bmod p$ for each σ_i.

Figure 3.1: Semi-OT$_n^k$-I: k-out-of-n OT against semi-honest receiver

3.2 OT$_n^k$ with Unconditional Security of Receiver

In this section we introduce OT$_n^k$ with unconditional receiver's security. We provide two schemes which are secure against semi-honest receiver in the standard model and secure against malicious receiver in the random oracle model respectively.

3.2.1 The Scheme for Semi-Honest Receiver

The sender S has n secret messages $m_1, m_2, \ldots, m_n$. Assume that the message space is $\mathbb{G}_q$. The receiver R wants to get $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ without revealing any information about $\sigma_1, \sigma_2, \ldots, \sigma_k$. The protocol Semi-OT$_n^k$-I is depicted in Figure 3.1.

For system parameters, let g, h be two generators of $\mathbb{G}_q$ where $\log_g h$ is unknown to all, and $\mathbb{G}_q$ be the group with some description. These parameters

can be used repeatedly by all possible senders and receivers as long as the value $\log_g h$ is not revealed. Therefore, $(g, h, \mathbb{G}_q)$ are universal parameters.

The receiver R first constructs a k-degree polynomial $f_2(x)$ such that $f_2(i) = 0$ if and only if $i \in \{\sigma_1, \ldots, \sigma_k\}$. Then R chooses another random k-degree polynomial $f_1(x)$ to mask the chosen polynomial $f_2(x)$. The masked choices $A_0, A_1, \ldots, A_{k-1}$ are sent to the sender S.

When S receives these queries, he first computes $B_i = g^{f_1(i)} h^{f_2(i)}$ by computing $A_0 A_1^i \cdots A_{k-1}^{i^{k-1}} (gh)^{i^k} \bmod p$. Because of the random polynomial $f_1(x)$, S does not know which $f_2(i)$ is equal to zero, for $i = 1, 2, \ldots, n$. Then S treats B_i as the public key and encrypts each message m_i by the ElGamal cryptosystem. The encrypted messages $c_1, c_2, \ldots, c_n$ are sent to R.

For each $c_i, i \in \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, since $B_i = g^{f_1(i)} h^{f_2(i)} = g^{f_1(i)} h^0 = g^{f_2(i)}$, R can get these messages by the decryption of ElGamal cryptosystem with secret key $f_1(i)$. If $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, since R can not compute $(g^{f_1(i)} h^{f_2(i)})^{r_i}$ with the knowledge of $g^{r_i}, f_1(i)$ and $f_2(i)$ only, the message m_i is unknown to R.

Correctness. Let $c_i = (U_i, V_i)$, we can check that the chosen messages m_{σ_i}, $i = 1, 2, \ldots, k$, are computed as

$$\begin{aligned} V_{\sigma_i} / U_{\sigma_i}^{f_1(\sigma_i)} &= m_{\sigma_i} \cdot (g^{f_1(\sigma_i)} h^{f_2(\sigma_i)})^{r_{\sigma_i}} / g^{r_{\sigma_i} f_1(\sigma_i)} \\ &= m_{\sigma_i} \cdot (g^{f_1(\sigma_i)} \cdot 1)^{r_{\sigma_i}} / g^{r_{\sigma_i} f_1(\sigma_i)} \\ &= m_{\sigma_i}. \end{aligned}$$

Security analysis. We now prove the security of Semi-OT$_n^k$-I.

Theorem 1 *For scheme Semi-OT$_n^k$-I, R's choices are unconditionally secure.*

Proof 1 *For any tuple $(A_0, A_1, \ldots, A_{k-1})$ sent by R, we have a pair of tuples $(a'_0, a'_1, \ldots, a'_{k-1})$ and $(b'_0, b'_1, \ldots, b'_{k-1})$ to represent all possible choices $\sigma'_1, \sigma'_2, \ldots, \sigma'_k$ so that $A_i = g^{a'_i} h^{b'_i}$ for $i = 0, 1, \ldots, k-1$. Thus, the receiver R's choices are unconditionally secure.*

□

Theorem 2 *Scheme Semi-OT$_n^k$-I meets the sender's security requirement. If the DDH assumption holds and R is semi-honest, R gets no information about messages m_i for $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$.*

Proof 2 *We show that for all $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$, c_i's look random if the DDH assumption holds. Let $\mathcal{I} = \{1, 2, \ldots, n\} \backslash \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ be an ordered set. Assume that there is a polynomial-time distinguisher $\mathcal{D}$ distinguishes the following two distributions:*

- $C = (g, h, \{(g^{r_{i_j}}, (g^{f_1(i_j)} h^{f_2(i_j)})^{r_{i_j}})\}_{i_j \in \mathcal{I}, j=1,\ldots,n-k})$,
  where $r_{i_j} \in_R \mathbb{Z}_q$, and f_1 and f_2 are chosen as in the protocol;
- $X = (g, h, \{(g^{r_{i_j}}, X_j)\}_{i_j \in \mathcal{I}, j=1,\ldots,n-k})$,
  where $r_{i_j} \in_R \mathbb{Z}_q$ and $X_j \in_R \mathbb{G}_q$.

Then we can construct another PPTM $\mathcal{D}'$, which takes $\mathcal{D}$ as a sub-routine, to distinguish the following two distributions:

- $\tilde{Y}_1 = \{(g, h, g^a, h^a)\}_{\mathbb{G}_q}$, *where* g, h *are generators of* $\mathbb{G}_q$, *and* $a \in_R \mathbb{Z}_q$.

- $\tilde{Y}_2 = \{(g, h, g^a, g^b)\}_{\mathbb{G}_q}$, *where* g, h *are generators of* $\mathbb{G}_q$, *and* $a, b \in_R \mathbb{Z}_q$.

The difference between $(\tilde{Y}_1, \tilde{Y}_2)$ *and* (Y_1, Y_2) *is that* h *can't be 1 in* $\tilde{Y}_1$ *and* $\tilde{Y}_2$.

Machine $\mathcal{D}'$

Input: (g, h, v, w) *(either from* $\tilde{Y}_1$ *or* $\tilde{Y}_2$*)*

1. *Randomly select* $\sigma_1, \ldots, \sigma_k \in \{1, \ldots, n\}$.

2. *Choose polynomials* f_1 *and* f_2 *according to* σ_i*'s as in the protocol.*

3. *Randomly select* $l \in \{1, 2, \ldots, n-k\}$.

4. *Output* $\mathcal{D}((U_{i_1}, V_{i_1}), (U_{i_2}, V_{i_2}), \ldots, (U_{i_{n-k}}, V_{i_{n-k}}))$, *where* $i_j \in \mathcal{I}$,

$$(U_{i_j}, V_{i_j}) = \begin{cases} (g^{r_{i_j}}, (g^{f_1(i_j)} h^{f_2(i_j)})^{r_{i_j}}) & \text{if } j \in \{1, \ldots, l-1\} \\ (v, v^{f_1(i_j)} w^{f_2(i_j)}) & \text{if } j = l \\ (g^{r_{i_j}}, X_j) & \text{if } j \in \{l+1, \ldots, n-k\} \end{cases},$$

and $r_{i_j} \in_R \mathbb{Z}_q, X_j \in_R \mathbb{G}_q$.

Assume that $\mathcal{D}$ *distinguishes* C *and* X *with non-negligible advantage* ε. *Let* $\alpha = (g, h, v, w)$ *and* $\vec{C}_l = (g, h, \{(U_{i_j}, V_{i_j})\}_{i_j \in \mathcal{I}, j=1,\ldots,n-k})$ *where*

$$(U_{i_j}, V_{i_j}) = \begin{cases} (g^{r_{i_j}}, (g^{f_1(i_j)} h^{f_2(i_j)})^{r_{i_j}}) & \text{if } j \in \{1, \ldots, l\} \\ (g^{r_{i_j}}, X_j) & \text{if } j \in \{l+1, \ldots, n-k\} \end{cases},$$

and $r_{i_j} \in_R \mathbb{Z}_q$ *and* $X_j \in_R \mathbb{G}_q$. *Note that* $\vec{C}_{n-k} = C$ *and* $\vec{C}_0 = X$. *If* α *is chosen from* $\tilde{Y}_1$, *then*

$$\Pr_{\alpha \in \tilde{Y}_1}[\mathcal{D}'(\alpha) = 1] = \Pr[\mathcal{D}'(\tilde{Y}_1) = 1] = \frac{1}{n-k} \sum_{l=1}^{n-k} \Pr[\mathcal{D}(\vec{C}_l) = 1].$$

If α is chosen from $\tilde{Y}_2$, then

$$\Pr_{\alpha \in \tilde{Y}_2}[\mathcal{D}'(\alpha) = 1] = \Pr[\mathcal{D}'(\tilde{Y}_2) = 1] = \frac{1}{n-k} \sum_{l=0}^{n-k-1} \Pr[\mathcal{D}(\vec{C}_l) = 1].$$

Therefore, we have

$$\begin{array}{rl} & \Pr[\mathcal{D}'(\tilde{Y}_1) = 1] - \Pr[\mathcal{D}'(\tilde{Y}_2) = 1] \\ = & \frac{1}{n-k}(\sum_{l=1}^{n-k} \Pr[\mathcal{D}(\vec{C}_l) = 1] - \sum_{l^*=0}^{n-k-1} \Pr[\mathcal{D}(\vec{C}_l) = 1]) \\ = & \frac{1}{n-k}(\Pr[\mathcal{D}(\vec{C}_{n-k}) = 1] - \Pr[\mathcal{D}(\vec{C}_0) = 1]) \\ = & \frac{1}{n-k}(\Pr[\mathcal{D}(C) = 1] - \Pr[\mathcal{D}(X) = 1]) \\ \geq & \frac{\varepsilon}{n-k}. \end{array}$$

Moreover, since $dist(\tilde{Y}_1, Y_1) = 1/q$ and $dist(\tilde{Y}_2, Y_2) = 1/q$, we can solve the DDH problem with at least non-negligible advantage $\frac{\varepsilon}{n-k} - \frac{2}{q}$, which is a contradiction.

□

Complexity. The scheme uses two rounds (steps 2 and 4), the first round sends k messages and the second round sends $2n$ messages. For computation, R computes $3k$ and S computes $(k+3)n$ modular exponentiations.

3.2.2 The Scheme for Malicious Receiver

A malicious player may not follow the protocol dutifully. For example, a malicious R might send arbitrary A_i's in step 2. So we present the scheme Mal-OT$_n^k$-I which is provable secure against malicious receivers in the random oracle model. The scheme is depicted in Figure 3.2.

The generator g and group $\mathbb{G}_q$ of system parameters are defined as that in Semi-OT$_n^k$-I. Let $H_1 : \{0,1\}^* \to \mathbb{G}_q, H_2 : \mathbb{G}_q \to \{0,1\}^l$ be two collision-

- System parameters: $(g, H_1, H_2, \mathbb{G}_q)$;
- S hasmessages: $m_1, m_2, \ldots, m_n \in \{0,1\}^l$;
- R's choices: $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$;

1. R computes $h_{\sigma_j} = H_1(\sigma_j)$ and $A_j = h_{\sigma_j}^{r_j}$, where $r_j \in_R \mathbb{Z}_q^*$ and $j = 1, 2, \ldots, k$.
2. $R \longrightarrow S$: $A_1, A_2, \ldots, A_k$.
3. S chooses a random $x \in \mathbb{Z}_q^*$ and computes $D_j = A_j^x$, $h_i = H_1(i)$, and $c_i = m_i \oplus H_2(h_i^x)$, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, k$.
4. $S \longrightarrow R$: $D_1, D_2, \ldots, D_k$, $c_1, c_2, \ldots, c_n$
5. R computes $K_j = D_j^{r_j^{-1}}$ and gets $m_{\sigma_j} = c_{\sigma_j} \oplus H_2(K_j)$ for $j = 1, 2, \ldots, k$.

Figure 3.2: Mal-OT$_n^k$-I: k-out-of-n OT against malicious receiver

resistant hash functions. Let messages be of l-bit length. Assume that CT-CDH is hard under $\mathbb{G}_q$.

Correctness. We can check that the chosen messages m_{σ_j}, $j = 1, 2, \ldots, k$, are computed as

$$\begin{aligned} c_{\sigma_j} \oplus H_2(K_j) &= m_{\sigma_j} \oplus H_2(h_{\sigma_j}^x) \oplus H_2(h_{\sigma_j}^x) \\ &= m_{\sigma_j}. \end{aligned}$$

Security analysis. We assume that the random oracle model in this security analysis.

Theorem 3 *In Mal-OT$_n^k$-I, R's choices are unconditionally secure.*

Proof 3 *Since $\mathbb{G}_q$ is the group of prime order q, all elements except 1 in $\mathbb{G}_q$ are generators. Given a value $A \in \mathbb{G}_q$, for any $h_i = H_1(i)$, there is an r_i such that $A = h_i^{r_i}$. That is, A can be a masked value of any index. Thus the receiver's choices are unconditionally secure.*

□

Theorem 4 *Even if R is malicious, the scheme Mal-OT_n^k-I meets the requirement for the sender's security in the random oracle model.*

Proof 4 *Since we treat H_2 as a random oracle, the malicious R has to know $K_i = h_i^x$ in order to query the hash oracle to get $H_2(h_i^x)$. For each possible malicious R, we construct a simulator R^* in the ideal model such that the outputs of R and R^* are indistinguishable.*

R^* works as follows:

1. *R^* simulates R to obtain $A_1^*, A_2^*, \ldots, A_k^*$. When R queries H_1 on index i, we return a random h_i^* (consistent with the previous queries.)*
2. *R^* simulates S (externally without knowing m_i's) on inputs A_1^*, $A_2^*, \ldots,$ A_k^* to obtain x^*, $D_1^*, D_2^*, \ldots, D_k^*$.*
3. *R^* randomly chooses $c_1^*, c_2^*, \ldots, c_n^*$.*
4. *R^* simulates R on input $(D_1^*, D_2^*, \ldots, D_k^*, c_1^*, c_2^*, \ldots, c_n^*)$ and monitors the queries closely. If R queries H_2 on some $v_j = (h_j^*)^{x^*}$, R^* sends j to the TTP T to obtain m_j and returns $c_j^* \oplus m_j$ as the hash value $H_2((h_j^*)^{x^*})$, otherwise, returns a random value (consistent with previous queries).*
5. *Output $(A_1^*, A_2^*, \ldots, A_k^*, D_1^*, D_2^*, \ldots, D_k^*, c_1^*, c_2^* \ldots, c_n^*)$.*

If R obtains $k+1$ decryption keys, R^* does not know which k indices are really chosen by R. The simulation would fail. Therefore we show that R can obtain at most k decryption keys by assuming the hardness of chosen-target CDH problem. In the above simulation, if R queries H_1, we return a random value output by the target oracle. When R^* simulates S on input $A_1^*, A_2^*, \ldots, A_k^*$, we forward these queries to the helper oracle, and return the corresponding outputs. Finally, if R queries H_2 on legal v_{j_i} for all $1 \leq i \leq k+1$, we can output $k+1$ pairs (v_{j_i}, j_i), which contradicts to the CT-CDH assumption. Thus, R obtains at most k decryption keys.

Let $\sigma_1, \sigma_2, \ldots, \sigma_k$ be the k choices of R. For the queried legal v_{σ_j}'s, c_{σ_j} is consistent with the returned hash values, for $j = 1, 2, \ldots, k$. Since no other $(h_l^*)^{x^*}$, $l \neq \sigma_1, \sigma_2, \ldots, \sigma_k$, can be queried to the H_2 hash oracle, c_l has the right distribution (due to the random oracle model). Thus, the output distribution is indistinguishable from that of R.

□

Complexity. Mal-OT$_n^k$-I has two rounds. The first round sends k messages and the second round sends $n+k$ messages. For computation, R computes $2k$, and S computes $n+k$ modular exponentiations.

3.3 OT$_n^k$ with Unconditional Security of Sender

In this section we introduce OT$_n^k$ with unconditional sender's security. We first provide a *generic* construction so that we can apply some multiplica-

tively homomorphic encryption schemes to it. Then we propose a concrete scheme with more efficient round complexity. Both constructions are proved secure in the standard model.

3.3.1 The Generic Construction

We propose a simple k-out-of-n oblivious transfer framework such that it can be widely used in small-scale applications or cryptographic protocols. Any multiplicatively homomorphic encryption scheme whose plaintext space has a prime order can be applied to this framework. Therefore protocol designers can choose the existing encryption scheme to implement OT in their protocols.

Current OT solutions using additively homomorphic encryption [AIR01, BGN05, LL07] have some security issues. The main problem is that most additively homomorphic encryption schemes have a composite plaintext space order. Thus the scheme needs either a trusted party to generate public/secret keys or a zero-knowledge proof system to ensure that the public key is well formed. Moreover, direct use of these homomorphic encryption schemes is not secure [Cha04, LL07].

In addition to the generality, our scheme has the following properties.

- The query phase of our construction has only two rounds: the receiver sends $O(k)$ messages to the sender, and gets $O(n)$ messages back. So it is very efficient and practical to be used as a building block.

- The construction doesn't need any zero-knowledge proof system but still remains secure against malicious receivers. This avoids additional rounds or the use of random oracles in the security proofs.

- The receiver's privacy is preserved if the underlying encryption scheme is IND-CPA secure. No additional assumption is needed.

Multiplicatively Homomorphic Public Key Encryption. A public key multiplicatively homomorphic encryption scheme $\mathcal{E} = (G, E, D)$ is defined as follows. Let $\mathcal{M}$ be the plaintext space.

- Key generation: on input a security parameter λ, $G(\lambda)$ outputs a pair of public and secret keys (pk, sk).

- Encryption: on input a message $m \in \mathcal{M}$, $E_{pk}(m)$ outputs a ciphertext C.

- Decryption: on input a ciphertext C, $D_{sk}(C)$ outputs the message m.

- Multiplicative homomorphism: for ciphertexts $C_1 = E_{pk}(m_1), C_2 = E_{pk}(m_2)$ and $C = E_{pk}(m)$, we have

  1. $C_1 \cdot C_2 = E_{pk}(m_1 \cdot m_2)$;

  2. $C^r = E_{pk}(m^r)$ where r is a known constant.

Note that if $\mathcal{M}$ has a prime order, there exists a generator generating all group elements in $\mathcal{M}$.

IND-CPA Security. We define the security game for a public key encryption scheme $\mathcal{E} = (G, E, D)$ as follows.

Setup. The challenger computes $(pk, sk) \leftarrow G(\lambda)$ and outputs the public key pk.

Challenge. The adversary outputs two messages m_0 and m_1. The challenger chooses a random $b \in \{0, 1\}$ and returns C_b representing the encryption of m_b.

Guess. Finally, the adversary outputs its guess b' and wins the game if $b = b'$.

We define the advantage of an adversary $\mathcal{A}$ in attacking $\mathcal{E}$ as the function of the security parameter λ: $Adv_{\mathcal{E},\mathcal{A}}(\lambda) = |Pr[b = b'] - \frac{1}{2}|$.

Definition 1 (IND-CPA Security) *We say that a public key encryption scheme is IND-CPA secure if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage against the challenger in the above game.*

Our construction Gen-OT$_n^k$-II is depicted in Figure 3.3. Let $\mathcal{E}$ be a multiplicatively homomorphic encryption scheme and $\mathcal{M}$ be the corresponding message space. Note that the order of $\mathcal{M}$ needs to be prime. The sender S has $m_1, \ldots, m_n \in \mathcal{M}$ and the receiver R has choices $\sigma_1, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$. The scheme is divided into two phases: Setup Phase and Query Phase.

In the setup phase, R computes his public/secret key pair, and sends pk to S. Then S and R engage in a 'proof of decryption ability' protocol

- Let $\mathcal{E} = (G, E, D)$ be a multiplicatively homomorphic encryption scheme;
- S has messages: $m_1, m_2, \ldots, m_n \in \mathcal{M}$;
- R's choice: $\sigma_1, \sigma_2 \ldots \sigma_k \in \{1, 2, \ldots, n\}$;

Setup Phase

1. R computes $G(\lambda)$ to generate keys (pk, sk) and sends pk to S.
2. S chooses a random $m \in \mathcal{M}$, and sends $E_{pk}(m)$ to R.
3. R replies the decryption result m. S aborts if it is incorrect.

Query Phase

1. R computes $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} + x^k = (x - \sigma_1)(x - \sigma_2)\cdots(x - \sigma_k)$ under the appropriate group.
2. $R \longrightarrow S$: $(g, A_0, A_1, \ldots, A_{k-1})$, where $A_i = E_{pk}(g^{a_i})$ for all $i = 0, 1, \ldots, k-1$ and g is a generator of $\mathcal{M}$.
3. $S \longrightarrow R$: $C_1, C_2, \ldots, C_n$, where
$$C_i = E_{pk}(m_i) \cdot (A_0 \cdot A_1^i \cdots A_{k-1}^{i^{k-1}} \cdot E_{pk}(g^{i^k}))^{r_i}$$
and r_i is randomly chosen.
4. R decrypts C_{σ_i} and gets m_{σ_i} for all $i = 1, 2, \ldots, k$.

Figure 3.3: Gen-OT$_n^k$-II: Generic k-out-of-n OT

to prevent R from learning the encrypted messages that R cannot decrypt himself. Note that the setup phase may not be necessary if the protocol that takes this OT construction as a building block already has a key setup for the receiver.

In the query phase, R first sets up a k-degree polynomial f such that $f(\sigma_i) = 0$ for all $1 \leq i \leq k$. In order to let S compute f, he puts the coefficients on the exponent of a generator g, and sends them to S in the encrypted form. Then S computes the following equations for all $i \in \{1, 2, \ldots, n\}$ using

the homomorphic property:

$$C_i = E_{pk}(m_i \cdot (g^{f(i)})^{r_i}),$$

where r_i is randomly chosen. Note that S has to encrypt the k-th term of $g^{f(x)}$ with coefficient 1 by himself. This is the technique of ensuring that $f(x)$ is a non-zero polynomial. Thus only k points makes f zero. R gets k messages and $n-k$ random values.

Theorem 5 *For any malicious receiver, the scheme Gen-OT$_n^k$-II has information-theoretic sender's security.*

Proof 5 *For each malicious receiver R^*, we construct a simulator $\mathcal{S}_{R^*}$ in the ideal model such that the output distributions of R^* and $\mathcal{S}_{R^*}$ are identical. $\mathcal{S}_{R^*}$ works as follows.*

1. *Simulate R^* to obtain $(pk, g, A_0, A_1, \ldots, A_{k-1})$.*
2. *Repeatedly rewind R^* to Step 2 of the setup phase and send $A_i \cdot E_{pk}(h_i), h_i \in_R \mathcal{M}$, as challenge ciphertexts to get $g^{a_i}h_i$ back. Compute $g^{a_i} = g^{a_i}h_i/h_i$ for all $i \in \{0, 1, \ldots, k-1\}$.*
3. *For all $i \in \{1, 2, \ldots, n\}$, compute $g^{f(i)}$ using the values g^{a_i}'s and define the set $J = \{j | g^{f(j)} = 1\}, |J| \leq k$.*
4. *Send all $j \in J$ to the TTP and get m_j's back.*
5. *Send C_i to R^* for all $i \in \{1, 2, \ldots, n\}$:*

$$C_i = \begin{cases} E_{pk}(m_i) & \text{if } i \in J; \\ E_{pk}(r_i^*) & \text{otherwise}, r_i^* \in_R \mathcal{M}. \end{cases}$$

6. *Output* $(pk, E_{pk}(m), m, g, A_0, A_1, \ldots, A_{k-1}, C_1, C_2, \ldots, C_n)$

In Step 2 of the simulation, we use the self reducibility properties of the encryption scheme to map an encrypted message to an encryption of a random message. Then the challenge ciphertext will be indeed the encryption of a random message. After getting the returned answer, we can compute the original message by canceling the random factor. Therefore we can get all messages sent by R^* *and thus know* R^**'s choices by checking whether* $g^{f(i)} = 1$*. Moreover, since the coefficient of* x^k *in* $f(x)$ *is 1, we can make sure that* $f(x)$ *will not be a zero polynomial. For a nonzero* k*-degree polynomial, there are at most* k *indices such that* $f(i) = 0$*. So the number of* R^**'s choices is at most* k*, and thus we can send these indices to the TTP.*

Let's consider the output distribution of $\mathcal{S}_{R^*}$*. First,* $(pk, g, A_0, \ldots, A_{k-1})$ *are outputted by* R^**. As stated above, the challenge chiphertext is the encryption of a random message, so the challenge and response are distributed as in the real scheme. For all* $i \notin J$ *(i.e.* $f(i) \neq 0$*), the ciphertexts* C_i*'s would be encryptions of random messages, which are identically distributed as* C_i*'s in the real scheme. On the other hand, as long as* $i \in J$*, we can get the index* i *and query the TTP to obtain* m_i*. Therefore all* C_i*'s are distributed as them in the real scheme. The scheme can be perfectly simulated and the sender's security is information-theoretic.*

Theorem 6 *For any malicious sender, the scheme Gen-*OT_n^k*-II has receiver's security if the underlying encryption scheme* $\mathcal{E} = (G, E, D)$ *is IND-CPA se-*

cure.

Proof 6 *Let S^* be any possible malicious sender. We can see that the challenges of S^* in the setup phase are independent of R's choices in the query phase. Moreover, S^* cannot influence R's choices before or after R sends $(g, A_0, A_1, \ldots, A_{k-1})$. So we can prove the receiver's security by just considering the values $(g, A_0, A_1, \ldots, A_{k-1})$.*

Suppose a polynomial time adversary $\mathcal{A}$ breaks receiver's security of Gen-OT_n^k-II with non-negligible advantage $\varepsilon(\lambda)$. We can construct another polynomial time adversary $\mathcal{B}$ breaking the IND-CPA security of $\mathcal{E}$ with advantage $\varepsilon(\lambda)/k$. Given pk as input, algorithm $\mathcal{B}$ works as follows.

1. *$\mathcal{B}$ gives algorithm $\mathcal{A}$ the public key pk and a generator g.*
2. *$\mathcal{A}$ outputs $\{\sigma_{0,1}, \ldots, \sigma_{0,k}\}, \{\sigma_{1,1}, \ldots, \sigma_{1,k}\} \in \{1, 2, \ldots, n\}^k$.*
3. *$\mathcal{B}$ computes the polynomials*
   - $f_0(x) = a_{0,0} + a_{0,1}x + \cdots + a_{0,k-1}x^{k-1} + x^k = (x - \sigma_{0,1})(x - \sigma_{0,2}) \cdots (x - \sigma_{0,k})$ *and*
   - $f_1(x) = a_{1,0} + a_{1,1}x + \cdots + a_{1,k-1}x^{k-1} + x^k = (x - \sigma_{1,1})(x - \sigma_{1,2}) \cdots (x - \sigma_{1,k})$.
4. *$\mathcal{B}$ randomly chooses a number $l \in \{1, 2, \ldots, k\}$ and sends $(g^{a_{0,l}}, g^{a_{1,l}})$ to the challenger of $\mathcal{E}$.*
5. *The challenger outputs a ciphertext C.*

6. $\mathcal{B}$ *sends*

$$\vec{C_l} = (E_{pk}(g^{a_{0,0}}), \ldots, E_{pk}(g^{a_{0,l-1}}), C, E_{pk}(g^{a_{1,l+1}}), \ldots, E_{pk}(g^{a_{1,k-1}}))$$

to $\mathcal{A}$.

7. $\mathcal{B}$ *outputs* $\mathcal{A}$*'s guess* b.

By the hybrid argument (similar to the proof of Theorem 2), we can see that

$$\mathcal{ADV}_{\mathcal{B}}(\lambda) \geq \frac{1}{k}\mathcal{ADV}_{\mathcal{A}}(\lambda) = \frac{\varepsilon(\lambda)}{k}$$

So $\mathcal{B}$ *breaks* $\mathcal{E}$ *with non-negligible advantage* $\varepsilon(\lambda)/k$*, which is a contradiction. Gen-OT*$_n^k$*-II has receiver's security.*

3.3.2 Concrete Construction

Here we propose a concrete OT_n^k construction with unconditional sender's security. The scheme is more efficient than the generic one. It is extended from the 1-out-of-n OT under the same security condition provided by Naor and Pinkas [NP01].

We present the scheme in Figure 3.4. The main idea of this scheme is the same as Semi-OT_n^k-I. R first chooses two polynomials $f_1(x), f_2(x)$ and a random value b where $f_2(x)$ represents the choices, and $f_1(x)$ and b are used to mask $f_2(x)$. By the DDH assumption, $C_i = g^{a_i b} h^{a'_i}$ can't be distinguished from the random value when given g^{a_i} and g^b, for $i = 0, 1, \ldots, k-1$. Therefore the choices of R are computationally secure.

- System parameters: $(g, \mathbb{G}_q)$;
- S has messages: $m_1, m_2, \ldots, m_n \in \mathbb{G}_q$;
- R's choices: $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$;

1. R chooses a generator h of $\mathbb{G}_q$, a random $b \in \mathbb{Z}_q$, and two polynomials $f_1(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} + x^k$ and $f_2(x) = a'_0 + a'_1 x + \cdots + a'_{k-1}x^{k-1} + x^k$ where $a_0, a_1, \ldots, a_{k-1} \in_R \mathbb{Z}_q$ and $a'_0 + a'_1 x + \cdots + a'_{k-1}x^{k-1} + x^k \equiv (x - \sigma_1)(x - \sigma_2)\cdots(x - \sigma_k) \bmod q$. Let $(A_0, A_1, \ldots, A_{k-1}) = (g^{a_0}, g^{a_1}, \ldots, g^{a_{k-1}}), B = g^b, (C_0, C_1, \ldots, C_{k-1}) = (g^{a_0 b}h^{a'_0}, g^{a_1 b}h^{a'_1}, \ldots, g^{a_{k-1} b}h^{a'_{k-1}})$.
2. $R \longrightarrow S : (h, A_0, A_1, \ldots, A_{k-1}, B, C_0, C_1, \ldots, C_{k-1})$.
3. S chooses n random pairs $(r_1, s_1), (r_2, s_2), \cdots, (r_n, s_n)$ in $\mathbb{Z}_q$, and computes $c_i = (g^{f_1(i)r_i}g^{s_i}, (g^{f_1(i)b}h^{f_2(i)})^{r_i}(g^b)^{s_i} \oplus m_i) = (X_i^{r_i}g^{s_i}, Z_i^{r_i}B^{s_i} \oplus m_i)$ for $i = 1, 2, \ldots, n$, where $X_i = A_0 A_1^i \cdots A_{k-1}^{i^{k-1}} g^{i^k}, Z_i = C_0 C_1^i \cdots C_{k-1}^{i^{k-1}} (gh)^{i^k}$.
4. $S \longrightarrow R$: $c_1, c_2, \ldots, c_n$.
5. Let $c_i = (U_i, V_i)$. R computes $m_{\sigma_i} = U_{\sigma_i}^b \oplus V_{\sigma_i}$ for each σ_i.

Figure 3.4: Con-OT$_n^k$-II: concrete k-out-of-n OT with unconditional sender's security

Then S encrypts the messages by the similar technique of randomized reduction of DDH from [NR97, Sta96]. The receiver R uses the value b to decrypt the chosen messages, and gets no information about other messages.

Correctness. Let $c_i = (U_i, V_i)$, we can check that the chosen messages m_{σ_i}, $i = 1, 2, \ldots, k$, are computed as

$$\begin{aligned} U_{\sigma_i}^b \oplus V_{\sigma_i} &= (g^{f_1(\sigma_i)r_{\sigma_i}}g^{s_{\sigma_i}})^b \oplus (g^{f_1(\sigma_i)b}h^{f_2(\sigma_i)})^{r_{\sigma_i}}(g^b)^{s_{\sigma_i}} \oplus m_{\sigma_i} \\ &= g^{f_1(\sigma_i)br_{\sigma_i} + bs_{\sigma_i}} \oplus (g^{f_1(\sigma_i)b} \cdot 1)^{r_{\sigma_i}} g^{bs_{\sigma_i}} \oplus m_{\sigma_i} \\ &= g^{f_1(\sigma_i)br_{\sigma_i} + bs_{\sigma_i}} \oplus g^{f_1(\sigma_i)br_{\sigma_i} + bs_{\sigma_i}} \oplus m_{\sigma_i} \\ &= m_{\sigma_i}. \end{aligned}$$

Security analysis. We now prove the security of Con-OT$_n^k$-II.

Theorem 7 *For any malicious sender, the scheme Con-OT$_n^k$-II has receiver's security if the DDH assumption holds.*

Proof 7 *Suppose a polynomial time adversary $\mathcal{A}$ breaks receiver's security of Con-OT$_n^k$-II with non-negligible advantage ε. We can construct another polynomial time adversary $\mathcal{B}$ solving the DDH problem with advantage ε/k. Given a DDH tuple (g, u, v, w) as input, algorithm $\mathcal{B}$ works as follows.*

1. *$\mathcal{B}$ gives algorithm $\mathcal{A}$ the generator g.*
2. *$\mathcal{A}$ outputs $\{\sigma_{0,1}, \ldots, \sigma_{0,k}\}, \{\sigma_{1,1}, \ldots, \sigma_{1,k}\} \in \{1, 2, \ldots, n\}^k$.*
3. *$\mathcal{B}$ performs the following steps:*
   (a) *randomly choose a number $l \in \{0, 1, \ldots, k-1\}$ and $h \in \mathbb{G}_q$;*
   (b) *compute the polynomials*
   - $f_0'(x) = a_{0,0}' + a_{0,1}'x + \cdots + a_{0,k-1}'x^{k-1} + x^k = (x - \sigma_{0,1})(x - \sigma_{0,2}) \cdots (x - \sigma_{0,k})$ *and*
   - $f_1'(x) = a_{1,0}' + a_{1,1}'x + \cdots + a_{1,k-1}'x^{k-1} + x^k = (x - \sigma_{1,1})(x - \sigma_{1,2}) \cdots (x - \sigma_{1,k})$;

   (c) *randomly choose $b \in \{0, 1\}$;*
   (d) *perform $\mathcal{A}(h, A_0, A_1, \ldots, A_{k-1}, v, C_0, C_1, \ldots, C_{k-1})$ where*

$$(A_i, C_i) = \begin{cases} (g^{a_i}, v^{a_i} h^{a'_{b,i}}) & \text{if } i \in \{0, \ldots, l-1\} \\ (u, w h^{a'_{b,l}}) & \text{if } i = l \\ (g^{a_i}, R_i h^{a'_{b,l+1}}) & \text{if } i \in \{l+1, \ldots, k-1\} \end{cases},$$

   and $a_i \in_R \mathbb{Z}_q, R_i \in_R \mathbb{G}_q$.

4. $\mathcal{A}$ outputs a guess b'. $\mathcal{B}$ outputs 1 if $b' = b$ and 0 otherwise.

Let $\alpha = (g, u, v, w)$. *Define* $\vec{E}_l = (g, A_1, A_2, \ldots, A_k, g^b, C_1, C_2, \ldots, C_k)$, *where*

$$(A_i, C_i) = \begin{cases} (g^{a_i}, g^{a_i b} h^{a'_{b,i}}) & \text{if } i \in \{0, \ldots, l\} \\ (g^{a_i}, R_i h^{a'_{b,l+1}}) & \text{if } i \in \{l+1, \ldots, k-1\} \end{cases}$$

for some $a_i \in \mathbb{Z}_q$ *and* $b \in \mathbb{Z}_q$. *If* α *is chosen from* Y_1, *then*

$$\Pr_{\alpha \in Y_1}[\mathcal{B}(\alpha) = 1] = \Pr[\mathcal{B}(Y_1) = 1] = \frac{1}{k} \sum_{l=1}^{k} \Pr[\mathcal{A}(\vec{E}_l) = b].$$

If α *is chosen from* Y_2, *then*

$$\Pr_{\alpha \in Y_2}[\mathcal{B}(\alpha) = 1] = \Pr[\mathcal{B}(Y_2) = 1] = \frac{1}{k} \sum_{l=0}^{k-1} \Pr[\mathcal{A}(\vec{E}_l) = b].$$

Therefore, we have

$$\begin{aligned} \Pr[\mathcal{B}(Y_1) = 1] - \Pr[\mathcal{B}(Y_2) = b] &= \frac{1}{k}\left(\sum_{l=1}^{k} \Pr[\mathcal{A}(\vec{E}_l) = b] - \sum_{l=0}^{k-1} \Pr[\mathcal{A}(\vec{E}_l) = 1]\right) \\ &= \frac{1}{k}(\Pr[\mathcal{A}(\vec{E}_k) = b] - \Pr[\mathcal{A}(\vec{E}_0) = b]) \\ &\geq \frac{\varepsilon}{k}. \end{aligned}$$

So we can solve the DDH problem with at least non-negligible advantage $\frac{\varepsilon}{k}$, *which is a contradiction. Con-OT$_n^k$-II has receiver's security.*

□

Theorem 8 *The sender's security of Scheme Con-OT$_n^k$-II is unconditionally-secure. That is, any receiver R gets no information about messages m_i, $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$.*

Proof 8 *Since* $Z_i = C_0 C_1^i \cdots C_{k-1}^{i^{k-1}} (gh)^{i^k} = g^{f_1^*(i)b^*} h^{f_2^*(i)}$, *the degree of* $f_2^*(x)$ *is* k. *That is, there are at most* k $f_2^*(i)$*'s equal to 0. Then for any other* $f_2^*(i) \neq 0$, *we prove that* $Z_i^{r_i} B^{s_i}$ *is uniformly distributed in* $\mathbb{G}_q$.

Let $\mathcal{I} = \{i \in \{1, 2, \ldots, n\} | f_2^*(i) \neq 0\}$. *For any* $i \in \mathcal{I}$, *we let* $\tilde{a}_i = f_1^*(i) r_i + s_i$, *and* $e_i = \log_g h^{f_2^*(i)}$. *Then*

$$\begin{aligned} Z_i^{r_i} B^{s_i} &= (g^{f_1^*(i)b} h^{f_2^*(i)})^{r_i} g^{bs_i} \\ &= g^{(f_1^*(i)r_i + s_i)b} (h^{f_2^*(i)})^{r_i} \\ &= g^{\tilde{a}_i b + e_i r_i}. \end{aligned}$$

Therefore, $Z_i^{r_i} B^{s_i}$ *is uniformly distributed in* $\mathbb{G}_q$ *because* $e_i \neq 0$ *and* r_i *is uniformly distributed in* $\mathbb{Z}_q$.

□

Complexity. Con-OT$_n^k$-II has two rounds. The first round sends $2k + 2$ messages and the second round sends n messages. For computation, R computes $4k + 1$, and S computes $(2k + 6)n$ modular exponentiations.

3.4 OT with Adaptive Queries

The queries of R in Mal-OT$_n^k$-I can be adaptive. In Mal-OT$_n^k$-I, computing the commitments c_i's are independent of computing the keys D_i's. Therefore, our scheme Mal-OT$_n^k$-I is adaptive in nature and the number k of queries need not be prefixed. Our Adpt-OT$_n^k$ scheme is depicted in Figure 3.5.

The protocol consists of two phases: the commitment phase and the transfer phase. The sender S first commits the messages in the commitment

- System parameters: $(g, H_1, H_2, \mathbb{G}_q)$;
- S has messages: $m_1, m_2, \ldots, m_n \in \mathbb{G}_q$;
- R's choices: $\sigma_1, \sigma_2, \ldots, \sigma_k \in \{1, 2, \ldots, n\}$;

Commitment Phase

1. S computes $c_i = m_i \oplus H_2(h_i^x)$ for $i = 1, 2, \ldots, n$, where $h_i = H_1(i)$, and $x \in_R \mathbb{Z}_q^*$.
2. $S \longrightarrow R : c_1, c_2, \ldots, c_n$.

Transfer Phase

For each σ_j, $j = 1, 2, \ldots, k$, R and S execute the following steps:

1. R chooses a random $r_j \in \mathbb{Z}_q^*$ and computes $h_{\sigma_j} = H_1(\sigma_j), A_j = (h_{\sigma_j})^{r_j}$.
2. $R \longrightarrow S : A_j$.
3. $S \longrightarrow R : D_j = (A_j)^x$.
4. R computes $K_j = (D_j)^{r_j^{-1}}$ and gets $m_{\sigma_j} = c_{\sigma_j} \oplus H_2(K_j)$.

Figure 3.5: Adpt-OT$_n^k$: Adaptive OT$_n^k$

phase. In the transfer phase, R sends a query A_j to S at a time and obtains the corresponding key to decrypt the commitment c_j.

Correctness of the scheme follows that of Mal-OT$_n^k$-I.

Security analysis. The security proofs are almost the same as those for Mal-OT$_n^k$-I.

Theorem 9 *In Adpt-OT$_n^k$, R's choice are unconditionally secure.*

Proof 9 *Since $\mathbb{G}_q$ is the group of prime order q, all elements except 1 in $\mathbb{G}_q$ are generators. For any $A_j = h_j^{r_j}$ and any h_l, $l \neq j$, there is an r_l that satisfies $A_j = h_l^{r_l}$. That is, A_j can be a masked value of any index. The receiver's choices are unconditionally secure.*

□

Theorem 10 *In the Adpt-OT$_n^k$, let $m_1, m_2, \ldots, m_n$ be the messages committed in the commitment phase. For any receiver R, the number of messages which R can get is less than or equal to the number of R's queries in the transfer phase in the random oracle model.*

Proof 10 *For any possible R, we construct a simulator R^* in the ideal model such that the outputs of R and R^* are indistinguishable:*

1. *In the commitment phase, R^* randomly chooses $c_1^*, c_2^*, \ldots, c_n^* \in \mathbb{G}_q$ and $x^* \in \mathbb{Z}_q^*$.*

2. *In the transfer phase, R^* simulates R on input $(c_1^*, c_2^*, \ldots, c_n^*)$, and gets message queries. For each query A_j, R^* returns $(A_j)^x$ to R. If R queries H_1 on index i, R^* returns a random $h_i^* \in \mathbb{G}_q$. If R queries H_2 on some K_i where*

   - *$K_i = (h_i^*)^x$ for some i, R^* sends i to the TTP T to obtain m_i and returns $c_i^* \oplus m_i$.*
   - *$K_i \neq (h_i^*)^x$ for all h_i^* have been queried to H_1, R^* returns a random value, and puts $(K_i)^{x^{-1}}$ to the revocation list of H_1.*

   Note that R^* uses a table for maintaining consistency of queries for each oracle. Moreover, R^* will not choose the values in the revocation list of H_1 as the answer of H_1 queries.

3. *Output $(c_1^*, c_2^* \ldots, c_n^*, A_1^*, A_2^*, \ldots, A_k^*, D_1^*, D_2^*, \ldots, D_k^*)$. (We assume that R makes k queries in the transfer phase: $A_1^*, A_2^*, \ldots, A_k^*$.)*

Since R makes k queries in the transfer phase, we show that R gets at most k messages by showing that R obtains at most k decryption keys. In the above simulation, if R queries H_1, we return a random value output by the target oracle. When R queries A_i^*'s adaptively, we forward these queries to the helper oracle, and return the corresponding outputs. Finally, if R queries H_2 on legal $(h_i^*)^x$ for all $1 \leq i \leq k+1$, we can output $k+1$ pairs $((h_i^*)^x, i)$, which contradicts to the CT-CDH assumption. Thus, R obtains at most k decryption keys.

Let $\sigma_1, \sigma_2, \ldots, \sigma_k$ be the k choices of R. For the legal query $(h_{\sigma_j}^*)^x$, c_{σ_j} is consistent with the returned hash values, for $j = 1, 2, \ldots, k$. Since no other $(h_l^*)^{x^*}$, $l \neq \sigma_1, \sigma_2, \ldots, \sigma_k$, can be queried to the H_2 hash oracle, c_l has the right distribution (due to the random oracle model). Thus, the output distribution is indistinguishable from R's output.

□

Complexity. In the commitment phase, S needs n modular exponentiations for computing the commitments c_i's. In the transfer phase, R needs 2 modular exponentiations for computing the query and the chosen message. S needs one modular exponentiation for answering each R's query. The commitment phase is one-round and the transfer phase is two-round for each adaptive query.

Chapter 4

Conditional Oblivious Cast

In this chapter, we introduce the notion of COC and propose various conditional oblivious cast schemes.

4.1 Preliminaries

In this section, we define the notion of COC and introduce some backgrounds.

4.1.1 Definitions

First, we give the formal definitions of COC-I and COC-II.

Conditional Oblivious Cast: Type I. Informally speaking, a COC-I scheme for predicate Q (Q-COC-I) has the following three properties:

- Correctness: both of A and B get m from S if $Q(x, y) = 1$.
- Sender's security: A and B cannot get any information about m if $Q(x, y) = 0$.

- Receiver's security: B and S cannot get any information about x; A and S cannot get any information about y;

The definition for Q-COC-I is as follows:

Definition 2 (Q-COC-I) *Let k be the security parameter, and A, B and S be all polynomial-time probabilistic Turing machines (PPTMs). Let $\mathcal{T} \leftarrow \langle A, B, S\rangle(\mu)$ be the communication transcript among A, B and S with common input μ. We say that a three-party interactive system $\Pi = (A, B, S)$ is a secure Q-COC-I scheme if it satisfies the following requirements for some constant c:*

1. *Correctness: For any $x, y, m, \mu \in \{0,1\}^{k^c}$ with $Q(x, y) = 1$,*

$$\Pr[A(x, \mu, \mathcal{T}) = m \wedge B(y, \mu, \mathcal{T}) = m : \mathcal{T} \leftarrow \langle A(x), B(y), S(m)\rangle(\mu)] = 1.$$

2. *Sender's security: For any PPTM A', B' and any $x, y, m_0, m_1, \mu \in \{0,1\}^{k^c}$ with $Q(x, y) = 0$, A' and B' together cannot distinguish the following two distributions with non-negligible advantage:*

   - $V_0^{\Pi, A', B'} = (x, y, \mu, \mathcal{T} \leftarrow \langle A'(x), B'(y), S(m_0)\rangle(\mu))$;
   - $V_1^{\Pi, A', B'} = (x, y, \mu, \mathcal{T} \leftarrow \langle A'(x), B'(y), S(m_1)\rangle(\mu))$.

   That is, A' and B' cannot distinguish their interaction with $S(m_0)$ from the interaction with $S(m_1)$.

3. *Receiver's security: For any PPTM A', S' and any $x, y_0, y_1, m, \mu \in \{0,1\}^{k^c}$ with $Q(x, y_0) = Q(x, y_1)$, A' and S' together cannot distinguish the following two distributions with non-negligible advantage:*

   - $V_0^{\Pi, A', S'} = (x, m, \mu, \mathcal{T} \leftarrow \langle A'(x), B(y_0), S'(m)\rangle(\mu))$;
   - $V_1^{\Pi, A', S'} = (x, m, \mu, \mathcal{T} \leftarrow \langle A'(x), B(y_1), S'(m)\rangle(\mu))$.

   That is, no distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ has non-negligible advantage so that $(y_0, y_1, st) \leftarrow \mathcal{D}_1(x, m, \mu)$ and $b \leftarrow \mathcal{D}_2(st, \mathcal{T} \leftarrow \langle A'(x), B(y_b), S'(m)\rangle(\mu))$, where $b \in_R \{0,1\}$ and st is the state information. The argument for B' and S' is similar to this argument.

Conditional Oblivious Cast: Type II. In COC-II, the message sender S holds two messages m_0 and m_1. A Q-COC-II scheme needs to satisfy the following three properties:

- Correctness: both of A and B get m_1 from S if $Q(x, y) = 1$, and m_0 if $Q(x, y) = 0$.
- Sender's security: A and B get exactly one message from S.
- Receiver's security: B and S cannot get any information about x; A and S cannot get any information about y;

The definition for Q-COC-II is as follows.

Definition 3 (Q-COC-II) *Let k be the security parameter, and A, B and S be all PPTMs. Let $\mathcal{T} \leftarrow \langle A, B, S\rangle(\mu)$ be the communication transcript among*

A, B and S with common input μ. We say that a three-party interactive system $\Pi = (A, B, S)$ is a secure Q-COC-II scheme if it satisfies the following requirements for some constant c:

1. *Correctness:*

   (a) *For any $x, y, m_0, m_1, \mu \in \{0,1\}^{k^c}$ with $Q(x, y) = 0$,*

   $$\Pr[A(x, \mu, \mathcal{T}) = m_0 \wedge B(y, \mu, \mathcal{T}) = m_0 : \mathcal{T} \leftarrow \langle A(x), B(y), S(m_0, m_1)\rangle(\mu)] = 1.$$

   (b) *For any $x, y, m_0, m_1, \mu \in \{0,1\}^{k^c}$ with $Q(x, y) = 1$,*

   $$\Pr[A(x, \mu, \mathcal{T}) = m_1 \wedge B(y, \mu, \mathcal{T}) = m_1 : \mathcal{T} \leftarrow \langle A(x), B(y), S(m_0, m_1)\rangle(\mu)] = 1.$$

2. *Sender's security: For any PPTM A', B' and any $x, y, m_0, m_1, m_1', \mu \in \{0,1\}^{k^c}$ with $Q(x, y) = 0$, A' and B' together cannot distinguish the following two distributions with non-negligible advantage:*

   - $V_0^{\Pi, A', B'} = (x, y, \mu, \mathcal{T} \leftarrow \langle A'(x), B'(y), S(m_0, m_1)\rangle(\mu))$,
   - $V_1^{\Pi, A', B'} = (x, y, \mu, \mathcal{T} \leftarrow \langle A'(x), B'(y), S(m_0, m_1')\rangle(\mu))$.

   The similar requirement is met for $Q(x, y) = 1$.

3. *Receiver's security: For any PPTM A', S' and any $x, y_0, y_1, m_0, m_1, \mu \in \{0,1\}^{k^c}$ with $Q(x, y_0) = Q(x, y_1)$, A' and S' together cannot distinguish the following two distributions with non-negligible advantage:*

- $V_0^{\Pi,A',S'} = (x, m_0, m_1, \mu, \mathcal{T} \leftarrow \langle A'(x), B(y_0), S'(m_0, m_1)\rangle(\mu))$,
- $V_1^{\Pi,A',S'} = (x, m_0, m_1, \mu, \mathcal{T} \leftarrow \langle A'(x), B(y_1), S'(m_0, m_1)\rangle(\mu))$.

The argument for B' and S' is similar to this argument.

Adversary Models. We classify the adversarial parties according to their intentions.

Semi-honest versus malicious. A party is semi-honest (or passive, honest-but-curious) if it follows the scheme step by step, but tries to compute extra information from received messages. A party is malicious (or active) if it can deviate from the specified steps in any way in order to get extra information.

Collusive versus non-collusive. We say that any two parties (two receivers or one sender and one receiver) are collusive if they use their mutual secrets to compute extra information. On the other hand, the parties are non-collusive if no two parties collude against the third one.

For clarity and simplicity, we first assume that all parties in our COC schemes are semi-honest and non-collusive. Then, we provide some techniques to transform the schemes to be secure against malicious and collusive parties in Section 4.4.

Also, we list some common notations used in this chapter in Table 4.1 for quick reference.

S, A, B	a sender and two receivers
$\mathcal{E}, \mathcal{E}'$	two encryption schemes
(G, E, D)	key generation, encryption and decryption functions of $\mathcal{E}$
(G', E', D')	key generation, encryption and decryption functions of $\mathcal{E}'$
(pk_S, sk_S)	a pair of sender's public key and secret key
(pk_R, sk_R)	a pair of receivers' common public key and secret key
$\mathcal{M}$	message space of an encryption scheme
m	the sender's message
m_0, m_1's	sender's two messages (COC-II)
x, y	the secret of A and B, respectively
$x_{[i]}, y_{[i]}$	the ith bit of x and y, respectively
n	length of x and y

Table 4.1: Common notations for our COC constructions

4.1.2 Backgrounds

We introduce some useful tools in this subsection. Let $x \in_R S$ mean that x is chosen from S uniformly and independently. We use $|x|$ to denote the length (in bits) of x, and $x_{[i]}$ to denote the i-th bit of x.

Additively Homomorphic Encryption Schemes A public key encryption scheme (G, E, D) is additively homomorphic if there is an operation $\boxplus$ satisfying, for any m_1 and m_2,

$$D_{SK}(E_{PK}(m_1) \boxplus E_{PK}(m_2)) = D_{SK}(E_{PK}(m_1 + m_2)),$$

where $(PK, SK) \leftarrow G(1^k)$ is the public/secret key pair and k is the security parameter. Note that if c is a known constant, we have the operation $\boxdot$ for all additively homomorphic encryption schemes:

$$c \boxdot E_{PK}(m) = \underbrace{E_{PK}(m) \boxplus E_{PK}(m) \boxplus \cdots \boxplus E_{PK}(m)}_{c \text{ times}} = E_{PK}(cm).$$

We introduce two additively homomorphic encryption schemes.

The Paillier Encryption Scheme [Pai99]

- $G(1^k) = (p, q, N, \alpha, g)$, where $N = pq$ is a k-bit number, p and q are two large primes, g is an integer of order $\alpha N \bmod N^2$ for some integer α. Let $PK = (g, N)$ and $SK = \lambda(N) = \text{lcm}(p-1, q-1)$.
- $E_{PK}(m) = g^m r^N \bmod N^2$, where $m \in \mathbb{Z}_N$ and $r \in_R \mathbb{Z}_N$.
- $D_{SK}(c) = \frac{L(c^{\lambda(N)} \bmod N^2, N)}{L(g^{\lambda(N)} \bmod N^2, N)} \bmod N$, where $L(u, N) = \frac{u-1}{N}$.

For any $m_1, m_2 \in \mathbb{Z}_N$, let $c_1 = E_{PK}(m_1) = g^{m_1} r_1^N \bmod N^2$ and $c_2 = E_{PK}(m_2) = g^{m_2} r_2^N \bmod N^2$. The operation $c_1 \boxplus c_2 = c_1 \cdot c_2 \bmod N^2$ is additively homomorphic because

$$\begin{aligned} D_{SK}(c_1 \boxplus c_2) &= D_{SK}((g^{m_1} r_1^N) \cdot (g^{m_2} r_2^N) \bmod N^2) \\ &= D_{SK}((g^{m_1+m_2}(r_1 r_2)^N) \bmod N^2) \\ &= D_{SK}(E_{PK}(m_1 + m_2)). \end{aligned}$$

The BGN encryption scheme [BGN05]. We say that $\mathbb{G}$ is a bilinear group of finite order if there exist a group $\mathbb{G}_1$ of the same order and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ such that for any $g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}$,

$$e(g^a, h^b) = e(g, h)^{ab}.$$

Moreover, for any generator g of $\mathbb{G}$, $e(g, g)$ should be a generator of $\mathbb{G}_1$.

The BGN encryption scheme is as follows.

- $G(1^k)$: let $N = pq$, where $|N| = k$ and p and q are two large primes. Construct a bilinear group $\mathbb{G}$ of order N along with the group $\mathbb{G}_1$ and

the bilinear map e. Let g, g' be two randomly chosen generators of $\mathbb{G}$, and $h = g'^q$, where is of order p. Output $PK = (N, \mathbb{G}, \mathbb{G}_1, e, g, h)$ and $SK = p$.

- $E_{PK}(m)$: let the message space $\mathcal{M} = \mathbb{Z}_T$, where $T << q$. Output $c = g^m h^r \in \mathbb{G}$, where $m \in \mathbb{M}$ and $r \in_R \mathbb{Z}_N$.

- $D_{SK}(c)$: if $c \in \mathbb{G}$, output $m = \log_{\hat{g}} c^p$ where $\hat{g} = g^p$. If $c \in \mathbb{G}_1$, output $m = \log_{\tilde{g}} c^p$ where $\tilde{g} = e(g, g)^p$.

Note that the decryption takes polynomial time in the size of the message space $\mathcal{M}$. The BGN scheme can only be used to encrypt short messages.

The BGN encryption scheme is not only additively homomorphic, but also one-time multiplicatively homomorphic. For any $m_1, m_2 \in \mathcal{M}$, let $c_1 = E_{PK}(m_1) = g^{m_1} h^{r_1} \in \mathbb{G}$ and $c_2 = E_{PK}(m_2) = g^{m_2} h^{r_2} \in \mathbb{G}$. The operation $c_1 \boxplus c_2 = c_1 \cdot c_2$ under group $\mathbb{G}$ is additively homomorphic because

$$\begin{aligned} D_{SK}(c_1 \boxplus c_2) &= D_{SK}(g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2}) \\ &= D_{SK}(g^{m_1+m_2} h^{r_1+r_2}) \\ &= D_{SK}(E_{PK}(m_1 + m_2)). \end{aligned}$$

The additive homomorphism also holds for $c_1, c_2 \in \mathbb{G}_1$.

On the other hand, consider the ciphertexts $c_1, c_2 \in \mathbb{G}$ defined above. Let $\tilde{g} = e(g, g) \in \mathbb{G}_1, \tilde{h} = e(g, h) \in \mathbb{G}_1$ and $h = g^\alpha$ for some $\alpha \in \mathbb{Z}_N$. The operation $c_1 \boxtimes c_2 = e(c_1, c_2) \in \mathbb{G}_1$ is one-time multiplicatively homomorphic

because

$$\begin{aligned} D_{SK}(c_1 \boxtimes c_2) &= D_{SK}(e(g^{m_1}h^{r_1}, g^{m_2}h^{r_2})) \\ &= D_{SK}(e(g, g^{m_2}h^{r_2})^{m_1+\alpha r_1}) \\ &= D_{SK}(e(g, g^{m_1 m_2}h^{m_2 r_1 + m_1 r_2 + \alpha r_1 r_2})) \\ &= D_{SK}(e(g,g)^{m_1 m_2} e(g,h)^{m_2 r_1 + m_1 r_2 + \alpha r_1 r_2}) \\ &= D_{SK}(\tilde{g}^{m_1 m_2} \tilde{h}^{m_2 r_1 + m_1 r_2 + \alpha r_1 r_2}) \\ &= D_{SK}(E_{PK}(m_1 m_2)). \end{aligned}$$

An encryption scheme (G, E, D) is semantically secure against adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ if for any PK generated by G, the probability that $(m_0, m_1, st) \leftarrow \mathcal{A}_1(PK)$ and $b \leftarrow \mathcal{A}_2(st, E_{PK}(m_b))$, where $b \in_R \{0,1\}$ and st is the state information, is negligible. The Paillier and BGN encryption schemes are semantically secure if the Computational Composite Residuosity and Subgroup Decision assumptions hold, respectively [BGN05, Pai99].

0-encoding and 1-encoding. 0-encoding and 1-encoding are two types of encoding to reduce the secure two-party computation problem for "greater than" predicate to the problem for "set intersection" predicate [LT05]. Let $s = s_{[n]}s_{[n-1]}\ldots s_{[1]} \in \{0,1\}^n$ be a binary string of length n. The 0-encoding of s is

$$S_s^0 = \{s_{[n]}s_{[n-1]}\ldots s_{[i+1]}1 | s_{[i]} = 0, 1 \le i \le n\};$$

and 1-coding of s is

$$S_s^1 = \{s_{[n]}s_{[n-1]}\ldots s_{[i]} | s_{[i]} = 1, 1 \le i \le n\}.$$

For two binary strings x, y of the same length, we have that $x > y$ if and only if there is exact one common element in S_x^1 and S_y^0.

4.2 Conditional Oblivious Cast: Type I

We provide COC-I schemes for three basic predicates: "equality", "inequality", and "greater than". We assume that A, B and S are semi-honest and non-collusive in this section. A and B privately agree on a public/secret key pair (pk_R, sk_R) of the homomorphic encryption scheme $\mathcal{E} = (G, E, D)$ with message space $\mathcal{M}$. S chooses his own key pair (pk_S, sk_S) of another semantically secure encryption scheme $\mathcal{E}' = (G', E', D')$ where the message space of $\mathcal{E}'$ covers the ciphertext space of $\mathcal{E}$. All parties exchange messages via a public channel. The secrets x, y held by A and B are assumed n-bit long.

4.2.1 COC-I for "Equality" Predicate

To determine if $x = y$, we compute $x - y$ via the homomorphic encryption scheme. If $x-y = 0$, A and B get the message m; otherwise, they get nothing. The scheme EQ-COC-I is described in Figure 4.1. A and B encrypt their secrets by pk_R and pk_S, and send them to S. S computes $m + r(x - y)$ in the encrypted form for a randomly chosen r and sends it back to A and B. Thus, A and B get m if and only if $x = y$.

Theorem 11 *Assume that the involved parties are semi-honest and non-collusive. The EQ-COC-I scheme is correct and secure if the underlying encryption schemes are semantically secure.*

- Message sender S has a message $m \in \mathcal{M}$ and a key pair (pk_S, sk_S) of $\mathcal{E}' = (G', E', D')$.
- Receivers A and B have a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \{0,1\}^n$.

1. A and B send $E'_{pk_S}(E_{pk_R}(x))$ and $E'_{pk_S}(E_{pk_R}(y))$ to S respectively.
2. S decrypts the received messages to get $E_{pk_R}(x)$ and $E_{pk_R}(y)$, and computes
$$c = E_{pk_R}(m) \boxplus (r \boxdot (E_{pk_R}(x) \boxplus (-1 \boxdot E_{pk_R}(y))))$$
and sends it to A and B, where $r \in_R \mathcal{M}$.
3. A and B compute $\hat{m} = D_{sk_R}(c)$ and verify whether $\hat{m}$ is valid.

Figure 4.1: COC-I scheme for "Equality" predicate: EQ-COC-I

Proof 11 *The EQ-COC-I scheme meets the following requirements.*

Correctness. If $x = y$, A and B can get m by computing

$$\begin{aligned} D_{sk_R}(c) &= D_{sk_R}(E_{pk_R}(m) \boxplus (r \boxdot (E_{pk_R}(x) \boxplus (-1 \boxdot E_{pk_R}(y))))) \\ &= D_{sk_R}(E_{pk_R}(m) \boxplus (r \boxdot E_{pk_R}(0))) \\ &= D_{sk_R}(E_{pk_R}(m)) \\ &= m. \end{aligned}$$

Sender's Security. We can see that

$$c = E_{pk_R}(m) \boxplus (r \boxdot (E_{pk_R}(x) \boxplus (-1 \boxdot E_{pk_R}(y)))) = E_{pk_R}(m + r(x - y)),$$

where $r \in_R \mathcal{M}$. For any possible m', there is another $r' \in \mathcal{M}$ such that $c = E_{pk_R}(m' + r'(x - y))$. As long as $x \neq y$, m is unconditionally secure for both A and B.

Receiver's Security. Since we assume that the parties are non-collusive in this scheme, we discuss the receiver's security against the other receiver and the sender separately. Moreover, the positions of A and B are symmetric, so we only prove the security of B (against A and S). For any semi-honest

S', assume that there is a polynomial-time distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ distinguishing the two distributions with non-negligible probability ε:

- $V_0^{\Pi,S'} = (m, pk_R, pk_S, sk_S, E'_{pk_S}(E_{pk_R}(x)), E'_{pk_S}(E_{pk_R}(y_0)))$;
- $V_1^{\Pi,S'} = (m, pk_R, pk_S, sk_S, E'_{pk_S}(E_{pk_R}(x)), E'_{pk_S}(E_{pk_R}(y_1)))$.

Then we can construct another PPTM $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, which takes $(\mathcal{D}_1, \mathcal{D}_2)$ as sub-routines, to break the semantic security of $\mathcal{E}$ with the same probability. Let pk_R^* be the challenge public key of $\mathcal{E}$ that $\mathcal{A}$ would like to attack.

Machine $\mathcal{A}_1(pk_R^*)$

1. *Randomly choose $m \in \mathcal{M}$ and (pk_S, sk_S) of $\mathcal{E}'$.*
2. *Perform $(y_0, y_1, (m, pk_R^*, pk_S, sk_S)) \leftarrow \mathcal{D}_1(m, (pk_R^*, pk_S, sk_S))$.*
3. *Let $m_0^* \leftarrow y_0$ and $m_1^* \leftarrow y_1$.*
4. *Output $(m_0^*, m_1^*, (y_0, y_1, m, pk_R^*, pk_S, sk_S))$.*

Machine $\mathcal{A}_2((y_0, y_1, m, pk_R^*, pk_S, sk_S), E_{pk_R^*}(m_b^*))$

1. *Randomly choose $x \in \mathcal{M}$ so that $x \neq y_0$ and $x \neq y_1$.*
2. *Perform $\hat{b} \leftarrow \mathcal{D}_2((y_0, y_1, m, pk_R^*, pk_S, sk_S), E'_{pk_S}(E_{pk_R^*}(x)), E'_{pk_S}(E_{pk_R^*}(m_b^*)))$.*
3. *Output $\hat{b}$.*

Since $m_0^* = y_0, m_1^* = y_1$, if $\mathcal{D}$ outputs a correct guess between the encryptions of y_0 and y_1, $\mathcal{A}$ also outputs a correct guess between the encryptions of m_0^* and m_1^*. Therefore $\mathcal{A}$ breaks the semantic security of $\mathcal{E}$ with probability ε.

On the other hand, for any semi-honest A', assume that there is a polynomial-time distinguisher $\mathcal{D}' = (\mathcal{D}'_1, \mathcal{D}'_2)$ distinguishing the following two distributions with non-negligible probability ε:

- $V_0^{\Pi,A'} = (x, pk_R, sk_R, pk_S, E'_{pk_S}(E_{pk_R}(y_0)), c)$;
- $V_1^{\Pi,A'} = (x, pk_R, sk_R, pk_S, E'_{pk_S}(E_{pk_R}(y_1)), c)$.

Then we can construct another PPTM $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$, which takes $(\mathcal{D}'_1, \mathcal{D}'_2)$ as sub-routines, to break the semantic security of $\mathcal{E}'$ with the same probability. Let pk_S^* be the challenge public key of $\mathcal{E}'$ that $\mathcal{A}'$ would like to attack.

Machine $\mathcal{A}'_1(pk_S^*)$

1. *Randomly choose $x \in \mathcal{M}$ and (pk_R, sk_R) of $\mathcal{E}$.*
2. *Perform $(y_0, y_1, (x, pk_R, sk_R, pk_S^*)) \leftarrow \mathcal{D}'_1(x, (pk_R, sk_R, pk_S^*))$.*
3. *Compute $m_0^* \leftarrow E_{pk_R}(y_0); m_1^* \leftarrow E_{pk_R}(y_1)$.*
4. *Output $(m_0^*, m_1^*, (y_0, y_1, x, pk_R, sk_R, pk_S^*))$.*

Machine $\mathcal{A}'_2((y_0, y_1, x, pk_R, sk_R, pk_S^*), E'_{pk_S^*}(m_b^*))$

1. *Compute $c \leftarrow E_{pk_R}(\tilde{m})$ for some $\tilde{m} \in_R \mathcal{M}$.*
2. *Perform $\hat{b} \leftarrow \mathcal{D}'_2((y_0, y_1, x, pk_R, sk_R, pk_S^*), E'_{pk_S^*}(m_b^*), c)$.*
3. *Output $\hat{b}$.*

Since $x \neq y_0$ and $x \neq y_1$, c can be treated as the encryption of the value $m + r(x - y)$, where $r \in_R \mathcal{M}$. Therefore c is identically distributed as the

- Message sender S has a message $m \in \mathcal{M}$ and a key pair (pk_S, sk_S) of $\mathcal{E}' = (G', E', D')$.
- Receivers A and B have a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \{0,1\}^n$.

1. A and B send $E'_{pk_S}(E_{pk_R}(x_{[i]}))$ and $E'_{pk_S}(E_{pk_R}(y_{[i]}))$ to S respectively, $1 \le i \le n$.
2. For each $i \in \{1, 2, \ldots, n\}$, S decrypts the received messages to get $E_{pk_R}(x_{[i]})$ and $E_{pk_R}(y_{[i]})$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_{[i]} - y_{[i]}$, $d'_i = x_{[i]} + y_{[i]} - 1$.
   (b) $e_i = 2e_{i+1} + d_i$, where $e_{n+1} = 0$.
   (c) $v_i = m + r_i(e_i - d_i + d'_i)$, where $r_i \in_R \mathcal{M}$
3. S sends $E_{pk_R}(v)$ in a random order to A and B, where $v = \langle v_1, v_2, \ldots, v_n \rangle$.
4. A and B decrypt the received messages and identify the correct message if existent.

Figure 4.2: COC-I scheme for "Inequality" predicate: INE-COC-I

real one. Finally, $\mathcal{D}'$ outputs a correct guess with non-negligible probability ε, $\mathcal{A}'$ breaks the semantic security of $\mathcal{E}'$ with probability ε, too.

4.2.2 COC-I for "Inequality" Predicate

COC-I for the "inequality" predicate is more complicated than that for the "equality" predicate. A and B need to bit-wisely encrypt their secrets, and send them to S. The scheme is depicted in Figure 4.2. Without loss of generality, we assume that $|x| = |y|$ for the two secrets x and y. To encrypt a vector $v = \langle v_1, v_2, \ldots, v_n \rangle$, we write $E_{pk_R}(v) = \langle E_{pk_R}(v_1), E_{pk_R}(v_2), \ldots, E_{pk_R}(v_n) \rangle$.

In the scheme, $d_i = x_{[i]} - y_{[i]}$ and $d'_i = x_{[i]} - \bar{y}_{[i]}$ are 0, 1, or -1. If $x_{[i]} = y_{[i]}, d_i = 0$; otherwise, $d'_i = 0$. Let l be the leftmost different bit

between x and y, i.e. the largest i such that $d_i \neq 0$. We have $e_i = 0$ if $i > l$, $e_i \neq 0$ if $i < l$, and $e_i = d_i$ if $i = l$.

If $x \neq y$, the message m is embedded into the index i at which $x_{[i]}$ and $y_{[i]}$ are distinct. However, in order to avoid leaking information of the number of distinct bits, S masks m with random values on all indices except the index l. It leaves only one copy of m in v_i's:

- For $i = l$, since $e_l = d_l$ and $d'_l = x_{[l]} - \bar{y}_{[l]} = 0$, $(e_l - d_l + d'_l) = 0$. Therefore, $v_l = m$.
- For $1 \leq i < l$, v_i would be a random value because $e_i - d_i + d'_i = 2e_{i+1} + d'_i \neq 0$ and $r_i \in_R \mathcal{M}$.
- For $l < i \leq n$, v_i is also a random value because $e_i = d_i = 0$, $d'_i \neq 0$ and $r_i \in_R \mathcal{M}$.

Theorem 12 *Assume that the involved parties are semi-honest and non-collusive. The INE-COC-I scheme is correct and secure if the underlying encryption schemes are semantically secure.*

Proof 12 *The INE-COC-I scheme meets the following requirements.*

Correctness. Let l be the index of the first different bit of x and y (from the most significant bit). We see that $d_l = e_l = x_{[l]} - y_{[l]} = 1$ or -1, and $d'_l = x_{[l]} - \bar{y}_{[l]} = 0$. Therefore, $v_l = m + r_l(e_l - d_l + d'_l) = m + r_l \cdot 0 = m$. Thus, A and B get m from the permutation of the encryptions.

Sender's Security. *We see that if $x = y$, all d_i's and e_i's are 0, and all d'_i's are not 0 (in fact, $+1$ or -1). Thus, for each index i, we have $v_i = m + r_i(0 \pm 1) = m \pm r_i$. Since for any possible m', there exists an r'_i such that $v_i = m' + r'_i$, m is unconditionally secure to A and B.*

Receiver's Security. *We prove the security of B against A and S, respectively. For any semi-honest S', assume that there is a polynomial-time distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$. Without loss of generality (by the hybrid argument), let $\mathcal{D}_1$ outputs only one y and $\mathcal{D}_2$ distinguishes the interaction with $B(y)$ and the interaction with $B(y')$, where y' differs from y in one chosen bit only. That is, $\mathcal{D}$ distinguishes the two distributions with non-negligible probability ε:*

- $V_0^{\Pi,S'} = (m, pk_R, pk_S, sk_S, X, Y_0)$;
- $V_1^{\Pi,S'} = (m, pk_R, pk_S, sk_S, X, Y_1)$,

where

$$\begin{aligned} X &= (E'_{pk_S}(E_{pk_R}(x_{[1]})), \ldots, E'_{pk_S}(E_{pk_R}(x_{[n]}))) \\ Y_0 &= (E'_{pk_S}(E_{pk_R}(y_{[1]})), \ldots, E'_{pk_S}(E_{pk_R}(y_{[n]}))) \\ Y_1 &= (E'_{pk_S}(E_{pk_R}(y_{[1]})), \ldots, E'_{pk_S}(E_{pk_R}(y_{[j-1]})), E'_{pk_S}(E_{pk_R}(\bar{y}_{[j]})), \\ &\quad E'_{pk_S}(E_{pk_R}(y_{[j+1]})), \ldots, E'_{pk_S}(E_{pk_R}(y_{[n]}))) \end{aligned}$$

for some chosen j. We can construct another PPTM $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, which takes $(\mathcal{D}_1, \mathcal{D}_2)$ as sub-routines, to break the semantic security of $\mathcal{E}$ with the same probability. Let pk_R^* be the challenge public key of $\mathcal{E}$ that $\mathcal{A}$ would like to attack.

Machine $\mathcal{A}_1(pk_R^*)$

1. *Randomly choose* $m \in \mathcal{M}$ *and* (pk_S, sk_S) *of* $\mathcal{E}'$.

2. *Perform* $(y, j, (m, pk_R^*, pk_S, sk_S)) \leftarrow \mathcal{D}_1(m, (pk_R^*, pk_S, sk_S))$.

3. *Let* $m_0^* \leftarrow y_{[j]}$ *and* $m_1^* \leftarrow \bar{y}_{[j]}$.

4. *Output* $(m_0^*, m_1^*, (y, j, m, pk_R^*, pk_S, sk_S))$.

Machine $\mathcal{A}_2((y, j, m, pk_R^*, pk_S, sk_S), E_{pk_R^*}(m_b^*))$

1. *Randomly choose* $x \in \mathcal{M}$ *so that* $x_{[i]} \neq y_{[i]}$ *for some* $i \neq j$.

2. *Compute* $X = (E'_{pk_S}(E_{pk_R^*}(x_{[1]})), \ldots, E'_{pk_S}(E_{pk_R^*}(x_{[n]})))$ *and*
   $Y = (E'_{pk_S}(E_{pk_R^*}(y_{[1]})), \ldots, E'_{pk_S}(E_{pk_R^*}(y_{[j-1]})), E_{pk_R^*}(m_b^*),$
   $E'_{pk_S}(E_{pk_R^*}(y_{[j+1]})), \ldots, E'_{pk_S}(E_{pk_R^*}(y_{[n]})))$.

3. *Perform* $\hat{b} \leftarrow \mathcal{D}_2((y, j, m, pk_R^*, pk_S, sk_S), X, Y)$.

4. *Output* $\hat{b}$.

Since $m_0^* = y_{[j]}, m_1^* = \bar{y}_{[j]}$, *if* $\mathcal{D}$ *outputs a correct guess between the encryptions of* $y_{[j]}$ *and* $\bar{y}_{[j]}$, $\mathcal{A}$ *also outputs a correct guess between the encryptions of* m_0^* *and* m_1^*. *Therefore* $\mathcal{A}$ *breaks the semantic security of* $\mathcal{E}$ *with probability* ε.

On the other hand, for any semi-honest A', *assume that there is a polynomial-time distinguisher* $\mathcal{D}' = (\mathcal{D}'_1, \mathcal{D}'_2)$ *distinguishing the two distributions with non-negligible probability* ε:

- $V_0^{\Pi, A'} = (x, pk_R, sk_R, pk_S, Y_0, E_{pk_R}(v))$;

- $V_1^{\Pi,A'} = (x, pk_R, sk_R, pk_S, Y_1, E_{pk_R}(v))$,

where Y_0 *and* Y_1 *are defined as above. Then we can construct another PPTM* $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$, *which takes* $(\mathcal{D}'_1, \mathcal{D}'_2)$ *as sub-routines, to break the semantic security of* $\mathcal{E}'$ *with the same probability. Let* pk_S^* *be the challenge public key of* $\mathcal{E}'$ *that* $\mathcal{A}'$ *would like to attack.*

Machine $\mathcal{A}'_1(pk_S^*)$

1. *Randomly choose* $x \in \mathcal{M}$ *and* (pk_R, sk_R) *of* $\mathcal{E}$.
2. *Perform* $(y, j, (x, pk_R, sk_R, pk_S^*)) \leftarrow \mathcal{D}'_1(x, (pk_R, sk_R, pk_S^*))$.
3. *Compute* $m_0^* \leftarrow E_{pk_R}(y_{[j]}); m_1^* \leftarrow E_{pk_R}(\bar{y}_{[j]})$.
4. *Output* $(m_0^*, m_1^*, (y, j, x, pk_R, sk_R, pk_S^*))$.

Machine $\mathcal{A}'_2((y, j, x, pk_R, sk_R, pk_S^*), E'_{pk_S^*}(m_b^*))$

1. *Compute* $E_{pk_R}(v)$ *where* $v = (v_1, v_2, \ldots, v_n) \in_R \mathcal{M}^n$.
2. *Compute* $Y = (E'_{pk_S^*}(E_{pk_R}(y_{[1]})), \ldots, E'_{pk_S^*}(E_{pk_R}(y_{[j-1]})), E'_{pk_S^*}(m_b^*),$ $E'_{pk_S^*}(E_{pk_R}(y_{[j+1]})), \ldots, E'_{pk_S^*}(E_{pk_R}(y_{[n]})))$.
3. *Perform* $\hat{b} \leftarrow \mathcal{D}'_2((y, j, x, pk_R, sk_R, pk_S^*), Y, E_{pk_R}(v))$.
4. *Output* $\hat{b}$.

Since $\mathcal{D}$ *outputs a correct guess with non-negligible probability* ε, $\mathcal{A}$ *also breaks the semantic security of* $\mathcal{E}$ *with* ε.

- Message sender S has a message $m \in \mathcal{M}$ and a key pair (pk_S, sk_S) of $\mathcal{E}' = (G', E', D')$.
- Receivers A and B have a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \{0,1\}^n$.

1. A encodes x as S_x^1, and sends $E'_{pk_S}(E_{pk_R}(S_x^1[i]))$ to S, $1 \le i \le n$.
2. B encodes y as S_y^0, and sends $E'_{pk_S}(E_{pk_R}(S_y^0[i]))$ to S, $1 \le i \le n$.
3. S decrypts the received messages and computes

$$e_i = E_{pk_R}(m) \boxplus (r_i \boxdot (E_{pk_R}(S_x^1[i]) \boxplus (-1 \boxdot E_{pk_R}(S_y^0[i])))),$$

   where $r_i \in_R \mathcal{M}$, $1 \le i \le n$. S sends e_i's to A and B in a random order.
4. A and B search $\hat{m}_i = D_{sk_R}(e_i)$, $1 \le i \le n$, to identify the correct m if existent.

Figure 4.3: COC-I scheme for "Greater Than" predicate: GT-COC-I

4.2.3 COC-I for "Greater Than" Predicate

For the "greater than" predicate, we use the encoding methods mentioned in Section 4.1.2. Receiver A encodes x via 1-encoding and receiver B encodes y via 0-encoding. The problem is then reduced to the "equality" problem immediately. When S receives encrypted S_x^1 and S_y^0, he checks equality for corresponding strings. Note that without loss of generality, we assume that S_x^1 and S_y^0 are ordered sets of n elements. If there is no element of a certain length in the sets, we fill it with a random element. The scheme is presented in Figure 4.3. The security argument is the same as that of the EQ-COC-I scheme. This method is more efficient than the GT-COC-II scheme (in the next section, by setting m_0 as a random number). We use $S[i]$ to denote the i-th element of the ordered set S here.

4.3 Conditional Oblivious Cast: Type II

In this section, we present COC-II schemes for the "equality" ("inequality") and the "greater than" predicates. We assume that A, B and S are semi-honest and non-collusive. A and B privately agree on a public/secret key pair (pk_R, sk_R) of the homomorphic encryption scheme $\mathcal{E} = (G, E, D)$ with message space $\mathcal{M}$. S chooses his own key pair (pk_S, sk_S) of another semantically secure encryption scheme $\mathcal{E}' = (G', E', D')$ where the message space of $\mathcal{E}'$ covers the ciphertext space of $\mathcal{E}$. All parties exchange messages via a public channel.

4.3.1 COC-II for "Equality" Predicate

Our COC-II scheme for the equality predicate is naturally extended from the EQ-COC-I and INE-COC-I schemes. Intuitively, if $x = y$, A and B get m_1 by the EQ-COC-I scheme and, otherwise, they get m_0 by the INE-COC-I scheme. The scheme is shown in Figure 4.4. It is almost the same as the INE-COC-I scheme except that S sends an extra ciphertext v_{eq} to A and B.

Theorem 13 *Assume that the involved parties are semi-honest and non-collusive. The EQ-COC-II scheme is correct and secure if the underlying encryption schemes are semantically secure.*

Proof 13 *We see that if $x = y$, all d_i's are equal to 0, and v_{eq} is equal to m_1. The opposite case holds by the same argument in the proof of Theorem 12. This ensures the correctness property.*

- Message sender S has two messages $m_0, m_1 \in \mathcal{M}$ and a key pair (pk_S, sk_S) of $\mathcal{E}' = (G', E', D')$.
- Receivers A and B have a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \{0,1\}^n$.

1. A and B send $E'_{pk_S}(E_{pk_R}(x_{[i]}))$ and $E'_{pk_S}(E_{pk_R}(y_{[i]}))$ to S respectively, $1 \leq i \leq n$.
2. For each $i \in \{1, 2, \ldots, n\}$, S decrypts the received messages to get $E_{pk_R}(x_{[i]})$ and $E_{pk_R}(y_{[i]})$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_{[i]} - y_{[i]}, d'_i = x_{[i]} + y_{[i]} - 1$.
   (b) $e_i = 2e_{i+1} + d_i$, where $e_{n+1} = 0$.
   (c) $v_{eq} = m_1 + \sum_{i=1}^{n} r_i d_i$, $v'_i = m_0 + r'_i(e_i - d_i + d'_i)$, where $r_i, r'_i \in_R \mathcal{M}$
3. S sends $E_{pk_R}(v_{eq}), E_{pk_R}(v')$ to A and B in a random order, where $v' = \langle v'_1, v'_2, \ldots, v'_n \rangle$.
4. A and B decrypt the received messages and identify the correct message.

Figure 4.4: COC-II scheme for "Equality" predicate: EQ-COC-II

For sender's security, let $r = \sum_{i=1}^{n} r_i d_i$. *Since* $r_i \in_R \mathcal{M}$, *if* $x \neq y$, *there is a* $d_i \neq 0$ *such that* r *is uniformly distributed, and thus* m_1 *is unconditionally secure to* A *and* B. *If* $x = y$, *by the proof of Theorem 12,* m_0 *is unconditionally secure to* A *and* B.

For receiver's security, S *gets no information about* x *and* y *by the semantic security of* $\mathcal{E}$. *For each of* A *and* B, *the secret is guaranteed by the semantic security of* $\mathcal{E}'$. *The proof can be found in the proof of Theorem 12. The receiver's security holds.*

4.3.2 COC-II for "Greater Than" Predicate

We can apply the GT-COC-I scheme twice to achieve a GT-COC-II scheme. One invocation is for testing $x > y$ and the other one is for testing $x \leq y$. But,

- Message sender S has two messages $m_0, m_1 \in \mathcal{M}$ and a key pair (pk_S, sk_S) of $\mathcal{E}' = (G', E', D')$.
- Receivers A and B have a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.
- Receiver A has a secret x, and receiver B has a secret y, where $x, y \in \{0,1\}^n$.

1. A and B send $E'_{pk_S}(E_{pk_R}(x_{[i]}))$ and $E'_{pk_S}(E_{pk_R}(y_{[i]}))$ to S respectively, $1 \le i \le n$.
2. For each $i \in \{1, 2, \ldots, n\}$, S decrypts the received messages to get $E_{pk_R}(x_{[i]})$ and $E_{pk_R}(y_{[i]})$, and computes the following values via homomorphic encryption:
   (a) $d_i = x_{[i]} - y_{[i]}, d'_i = x_{[i]} + y_{[i]} - 1$.
   (b) $e_i = r_i e_{i+1} + d_i$, $e'_i = r'_i d'_i$, where $e_{n+1} = 0$, $r_i, r'_i \in_R \mathcal{M}$
   (c) $f_i = e_i + e'_i$
   (d) $v_i = \frac{m_1 - m_0}{2} f_i + \frac{m_1 + m_0}{2}$, $v_{eq} = m_0 + \sum_{i=1}^{n} r''_i d_i$, where $r''_i \in_R \mathcal{M}$.
3. S sends $E_{pk_R}(v), E_{pk_R}(v_{eq})$ in a random order to A and B, where $v = \langle v_1, v_2, \ldots, v_n \rangle$.
4. A and B decrypt the received messages and identify the correct message.

Figure 4.5: COC-II scheme for "Greater Than" predicate: GT-COC-II

this approach costs twice as much as the GT-COC-I scheme. Our scheme for GT-COC-II in Figure 4.5 is more efficient. It uses an extra ciphertext (for the case $x = y$) from S to A and B only.

Let l be the leftmost different bit between x and y. For $i < l$ and $i > l$, e_i and e'_i would be random values in $\mathcal{M}$, respectively. When $i = l$, we have $e_i = d_i$ and $e'_i = 0$. Therefore, f_i is a random value when $i \neq l$ and $f_l = d_l$. If $x > y$, $f_l = 1$ and thus $c_l = m_1$; if $x < y$, $f_l = -1$ and thus $v_l = m_0$. For the case $x = y$, we use an extra value v_{eq} to embed m_0 like scheme EQ-COC-II.

Theorem 14 *Assume that the involved parties are semi-honest and non-collusive. The GT-COC-II scheme is correct and secure if the underlying*

homomorphic encryption schemes are semantically secure.

Proof 14 *The GT-COC-II scheme meets the following requirements.*

Correctness. Consider the following three cases:

- $x > y$*: let* l *be the index of the first different bit of* x *and* y *(from the most significant bit), we have* $e_l = d_l = 1, e'_l = d'_l = 0$*, and thus* $f_l = e_l + e'_l = 1$*. Therefore* $v_l = \frac{m_1 - m_0}{2} \cdot 1 + \frac{m_1 + m_0}{2} = m_1$*.*

- $x < y$*: similarly, since* $f_l = e_l = d_l = -1$ *in this case, we have* $v_l = \frac{m_1 - m_0}{2} \cdot (-1) + \frac{m_1 + m_0}{2} = m_0$*.*

- $x = y$*: by the same argument in the proof of Theorem 13,* A *and* B *get* m_0 *from* v_{eq}*.*

Sender's Security. We see that if $x \neq y$*, then for all* $i \neq l$*,* f_i *is uniformly distributed in* $\mathcal{M}$*. That is, all* v_i*'s except* v_l *are uniformly distributed in* $\mathcal{M}$*. For index* l*, according to the above argument,* $v_l = m_0$ *if* $x < y$ *and* $v_l = m_1$ *if* $x > y$*. Moreover, by the proof of Theorem 13,* $v_{eq} = m_0$ *if* $x = y$*, and* v_{eq} *is uniformly distributed if* $x \neq y$*. Therefore,* m_0 *is unconditionally secure to* A *and* B *if* $x > y$*, and* m_1 *is unconditionally secure to* A *and* B *if* $x \leq y$*.*

Receiver's Security. Since the receivers encrypt their secrets in the same way as in the previous schemes, the proof is also the same as the proof of Theorem 12.

4.3.3 A General Transformation from COC-II to COC-III

Based on the COC-II schemes, we can construct COC-III schemes for the corresponding predicates. Assume that we have a COC-II scheme for some predicate Q. Then we can construct the Q-COC-III by the following steps:

1. A and B choose their own public/secret key pairs, namely, (pk_A, sk_A) and (pk_B, sk_B).
2. S lets $m_1 = E_{pk_A}(m)$ and $m_0 = E_{pk_B}(m)$.
3. All parties perform Q-COC-II as usual.
4. A and B then decrypt the received message by their own secret keys.

We see that if Q holds, both A and B get $m_1 = E_{pk_A}(m)$. But, only A can decrypt it to get the message m. Similarly, if Q does not hold, only B gets the message.

4.4 COC Against Malicious Parties and Collusion

In this section we show how to modify our COC schemes against malicious parties and their collusion. Since all parties should be able to check whether the computations are performed correctly, we assume that the existence of a bulletin board such that all parties can post the encryptions of their secrets onto the board, and publicly perform their computations. Furthermore, all

parties should perform the decryption jointly such that no collusive parties can get the secret information of the other party. We first introduce some building blocks. Then we integrate these building blocks for a secure INE-COC-I scheme against malicious parties. Here we take the Paillier encryption scheme as the example to describe these sub-functions. For other encryption schemes, there exist similar constructions.

The Distributed Cryptosystem. Because all parties post the encryptions of their secrets on the bulletin board, we need a distributed version of the cryptosystem so that no (collusive) parties can decrypt messages without the agreement of *all* parties. Assume that each party gets a key share in the setup phase (from a dealer or a distributed key generation protocol). To decrypt a ciphertext, all parties output their partial decryptions. They can get the message from the combination of these partial decryptions. Fouque et al. [FPS00] provided a threshold Paillier encryption scheme. We have the following lemma from their work.

Lemma 1 *Under the decisional composite residuosity assumption and the random oracle model, there is a threshold Paillier cryptosystem which is semantically secure against active non-adaptive adversaries.*

Knowledge Proof Systems. We need some knowledge proof systems to verify the correctness of parties' computations. There exist some interactive proof systems for the Paillier encryption scheme. We can make them

non-interactive by using the Fiat-Shamir heuristic [FS86]. We provide the following sub-functions (assuming pk is the common public key):

- NI-PPK($E_{pk}(m)$) (Non-Interactive Proof of Plaintext Knowledge):

  First of all, all parties should commit their secrets. They post the encryptions of their secrets along with non-interactive proofs of knowledge of their secrets. The prover proves that he knows the plaintext m for the encryption $E_{pk}(m)$. Cramer et al. [CDN01] provided such an interactive proof system.

- NI-PEB($E_{PK}(b)$) (Non-Interactive Proof of Encryption of a Bit):

  In some schemes, A and B commit their "bitwise" secrets rather than the whole secrets. So all parties should be able to check that a ciphertext $E_{pk}(b)$ is indeed the encryption of a bit. Baudron [BS01] et al. introduced such an interactive proof system.

- NI-PCM($E_{pk}(r), E_{pk}(m), E_{pk}(rm)$) (Non-Interactive Proof of Correct Multiplication):

  In our schemes, all parties need to do multiplication on a known constant and a ciphertext. However, the random constant could not be known by others. So each party posts the encryption of the constant $E_{pk}(r)$ and the result of multiplication $E_{pk}(rm)$. All parties can check that $E_{pk}(rm)$ is indeed the encryption of rm. Cramer et al. [CDN01] provided this interactive proof system.

By the respective works, we have the following lemma.

Lemma 2 *In the random oracle model, there exist NI-PPK, NI-PEK, and NI-PCM zero-knowledge proof systems.*

The Mix-Net System. Mix-net is a cryptographic system providing anonymous and unlinkable communication. It consists of a set of servers that shuffle a list of ciphertexts so that ciphertexts in the output list cannot be linked to those in the input list. To ensure correct shuffling, the output list should be verified that it is indeed a permutation of the input list. We use Mix-Net$(\cdot)$ to denote the mix-net sub-protocol which outputs the list of shuffled input ciphertexts. We can find such a mix-net system for the Paillier encryption scheme in the work of Nguyen et al. [NSNK04]. Also, we have the following lemma from their work.

Lemma 3 *There is a Mix-Net sub-protocol which provides indistinguishability under chosen permutation attack if the decisional composite residuosity assumption holds.*

Putting Things Together. Now we take the INE-COC-I scheme as an example to show how to build a scheme secure against malicious parties by the above tools. In the initial stage, A, B and S get their shares of the secret key corresponding to pk. We present the new protocol in Figure 4.6.

At the beginning of the protocol, all parties post the encryption of their secrets on the bulletin board with the corresponding NI-PPK proofs. In

addition, A and B provide the NI-PEB proofs to convince others of their bitwise secrets. Also, S posts the encryptions of constants 0 and 1 for the later computation and provides the random factors used in the encryption such that all parties can check the correctness. Then all parties perform the computation as S performs in the INE-COC-I scheme. Since the random r_i's cannot be known by any party (we need to consider the collusion of the sender and a receiver), all parties need to generate these r_i's jointly. For each $1 \leq i \leq n$, they first generate their own random values r_{Ai}, r_{Bi} and r_{Si} respectively, and perform the multiplication by themselves. The corresponding NI-PCM proofs are also posted for verification. Then they sum up their computation results, and the new random value is implicitly defined as $r_{Ai}+r_{Bi}+r_{Si}$. After that, all parties execute the Mix-Net system to get the shuffled ciphertexts. Finally, S sends the partial decryptions of these ciphertexts to A and B, and A and B also exchange their partial decryptions. With these partial decryptions, A and B can decrypt the ciphertexts.

Theorem 15 *The MAL-INE-COC-I scheme is correct and secure if the decisional composite residuosity assumption holds.*

Proof 15 *Based on security assumption of the Paillier encryption scheme, the MAL-INE-COC-I scheme meets the following requirements.*

Correctness. By the NI-PEB proofs, we know that $E_{x_{[i]}}$'s and $E_{y_{[i]}}$'s are fair encryptions of bits. Let l be the index of the first different bit of x and y (from the most significant bit). We see that $D_{sk}(E_{d_l}) = D_{sk}(E_{e_l}) = x_{[l]} - y_{[l]} =$

1 *or* -1, *and* $D_{sk}(E_{d'_l}) = x_{[l]} - \bar{y}_{[l]} = 0$. *Therefore, we have* $D_{sk}(E_{f'_l}) = 0$ *and* $D_{sk}(E_{v_l}) = m + (r_{Al} + r_{Bl} + r_{Sl}) \cdot 0 = m$. *Moreover, each* $E_{v'_i}$ *is just the permutation and reencryption of some* E_{v_j}. *A and B can get m from these encryptions.*

Sender's Security. For any malicious A' *and* B', *we construct a simulator* $\mathcal{SIM}_1$ *to simulate the view of the adversary* $\mathcal{ADV}_1$ *who controls* A' *and* B'. *Let* pk^* *be the challenge public key that* $\mathcal{ADV}_1$ *would like to attack. The simulator* $\mathcal{SIM}_1$ *randomly chooses the secret key shares for* A' *and* B', *named* sk_A *and* sk_B, *respectively. On input* (pk^*, sk_A, sk_B), $\mathcal{ADV}_1$ *first outputs two messages* m_0^* *and* m_1^*. *We show that if* $\mathcal{ADV}_1$ *distinguishes between the interaction with* $S(m_0^*)$ *and the interaction with* $S(m_1^*)$ *with non-negligible probability, then we can break Lemma 1, the semantic security of the threshold Paillier encryption scheme.*

Given the challenge ciphertext $E_{pk^*}(m_b^*)$ *for some* $b \in \{0, 1\}$, $\mathcal{SIM}_1$ *has to output its guess. It first posts* $E_m = E_{pk^*}(m_b^*)$ *on the board, and simulates the NI-PPK proof in the random oracle model.* $\mathcal{SIM}_1$ *and* $\mathcal{ADV}_1$ *then perform the subsequent steps as the real scheme until step 4(d). In step 4(d),* $\mathcal{SIM}_1$ *first randomly chooses* $\tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_n \in \mathbb{Z}_N$. *After* $\mathcal{ADV}_1$ *posts* $E_{f_{Ai}}$*'s and* $E_{f_{Bi}}$*'s,* $\mathcal{SIM}_1$ *computes and posts*

$$E_{f_{Si}} = E_{\tilde{v}_i} \boxplus (-1 \boxdot E_m) \boxplus (-1 \boxdot E_{f_{Ai}}) \boxplus (-1 \boxdot E_{f_{Bi}}),$$

where $E_{\tilde{v}_i} = E_{PK}(\tilde{m}_i), 1 \leq i \leq n$. $\mathcal{SIM}_1$ *also posts random ciphertexts* $E_{r_{Si}}$*'s and simulates the NI-PCM proof in the random oracle model. Then*

they continue running the scheme until step 6. In step 6, $\mathcal{SIM}_1$ *chooses a random permutation* π*, and outputs the partial decryptions such that* $\mathcal{ADV}_1$ *decrypts* $(E_{v'_1}, E_{v'_2}, \ldots, E_{v'_n})$ *to* $(\tilde{m}_{\pi(1)}, \tilde{m}_{\pi(2)}, \ldots, \tilde{m}_{\pi(n)})$*.*

We show that all data simulated by $\mathcal{SIM}_1$ *cannot be distinguished from the real ones by* $\mathcal{ADV}_1$*. First, since the key shares of the Paillier encryption scheme are uniformly distributed, the two shares given by* $\mathcal{SIM}_1$ *are indistinguishable from the real ones. By Lemma 2,* $\mathcal{SIM}_1$ *can simulate the NI-PPK and NI-PCM proofs in the random oracle model. The encryptions* $E_{f_{Si}}$*'s are distributed as the real ones because* $\mathcal{SIM}_1$ *computes them from the final results. Furthermore, by the proof of Theorem 12, if* $x = y$*,* E_{f_i}*'s must be uniformly distributed. So the encryptions* v_i*'s are distributed as the real ones. Finally, by Lemma 3,* $\mathcal{ADV}_1$ *cannot distinguish the permutation* π *from the real one.* $\mathcal{SIM}_1$ *successfully simulates the scheme. After that,* $\mathcal{ADV}_1$ *outputs the guess* $\hat{b}$*, and* $\mathcal{SIM}_1$ *also outputs* $\hat{b}$ *directly. If* $\hat{b} = b$ *with non-negligible probability,* $\mathcal{SIM}_1$ *breaks Lemma 1.*

Receiver's Security. We also construct a simulator $\mathcal{SIM}_2$ *to simulate the view of the adversary* $\mathcal{ADV}_2$ *who controls* A' *and* S'*. Let* pk^* *be the challenge public key that* $\mathcal{ADV}_2$ *would like to attack. The simulator* $\mathcal{SIM}_2$ *randomly chooses the secret key shares for* A' *and* S'*, named* sk_A *and* sk_S*, respectively. On input* (pk^*, sk_A, sk_S)*,* $\mathcal{ADV}_2$ *outputs a secret* y *and an index* j*. We show that if* $\mathcal{ADV}_2$ *distinguishes between the interaction with* $B(y_{[1]}y_{[2]}\ldots y_{[n]})$ *and the interaction with* $B(y_{[1]}\ldots y_{[j-1]}\bar{y}_{[j]}y_{[j+1]}\ldots y_{[n]})$ *with non-negligible prob-*

ability, then we can break Lemma 1, the semantic security of the threshold Paillier encryption scheme.

Let $y_0^* = y_{[1]}y_{[2]}\dots y_{[n]}$ *and* $y_1^* = y_{[1]}\dots y_{[j-1]}\bar{y}_{[j]}y_{[j+1]}\dots y_{[n]}$. *Given the challenge ciphertext* $E_{pk^*}(y_b^*)$ *for some* $b \in \{0,1\}$, $\mathcal{SIM}_2$ *has to output its guess. It first posts* $E_{y^*_{b[i]}} = E_{pk^*}(y^*_{b[i]})$ *for all* $1 \le i \le n$ *on the board, and simulates the corresponding NI-PPK and NI-PEB proof in the random oracle model. After* $\mathcal{ADV}_2$ *posts* E_m *and* $E_{x_{[i]}}$ *for all* i, $\mathcal{SIM}_2$ *extracts* m *and* y *in the corresponding proofs. Then* $\mathcal{SIM}_2$ *and* $\mathcal{ADV}_2$ *perform the subsequent steps as the real scheme until step 4(d). In step 4(d), if* $x = y$ $(Q(x,y) = 0)$, $\mathcal{SIM}_2$ *randomly chooses* $\tilde{m}_1, \tilde{m}_2, \dots, \tilde{m}_n \in \mathbb{Z}_N$; *otherwise* $(Q(x,y) = 1)$, $\mathcal{SIM}_2$ *sets* $\tilde{m}_1 = m$ *and randomly chooses* $\tilde{m}_2, \dots, \tilde{m}_n \in \mathbb{Z}_N$. *After* $\mathcal{ADV}_2$ *posts* $E_{f_{Si}}$*'s and* $E_{f_{Ai}}$*'s,* $\mathcal{SIM}_2$ *computes and posts*

$$E_{f_{Bi}} = E_{\tilde{v}_i} \boxplus (-1 \boxdot E_m) \boxplus (-1 \boxdot E_{f_{Si}}) \boxplus (-1 \boxdot E_{f_{Ai}}),$$

where $E_{\tilde{v}_i} = E_{PK}(\tilde{m}_i), 1 \le i \le n$. $\mathcal{SIM}_2$ *also posts random ciphertexts* $E_{r_{Bi}}$*'s and simulates the NI-PCM proof in the random oracle model. Then they continue running the scheme until step 6. In step 6,* $\mathcal{SIM}_2$ *chooses a random permutation* π, *and outputs the partial decryptions such that* $\mathcal{ADV}_2$ *decrypts* $(E_{v'_1}, E_{v'_2}, \dots, E_{v'_n})$ *to* $(\tilde{m}_{\pi(1)}, \tilde{m}_{\pi(2)}, \dots, \tilde{m}_{\pi(n)})$.

As the argument of sender's security, all data simulated by $\mathcal{SIM}_2$ *cannot be distinguished from the real ones by* $\mathcal{ADV}_2$. *So* $\mathcal{SIM}_2$ *successfully simulates the scheme. Finally,* $\mathcal{ADV}_2$ *outputs the guess* $\hat{b}$, *and* $\mathcal{SIM}_2$ *also outputs* $\hat{b}$ *directly. If* $\hat{b} = b$ *with non-negligible probability,* $\mathcal{SIM}_2$ *breaks Lemma 1.*

The security argument of A against the adversary who controls B and S is similar.

4.5 Extensions

We discuss some extensions of our COC schemes in this section. We provide some new solutions for three-party oblivious cast and consider COC constructions for other predicates.

4.5.1 Constructions of Oblivious Cast

As mentioned in Section 1, Blaze [Bla96] first introduces the notion of OC, and presents a k-out-of-n OC scheme based on blind signatures. Their OC scheme assumes an anonymous communication mechanism for hiding the identity of the sender. It needs four rounds to complete the protocol. Here we provide other solutions based on COC-III or homomorphic encryption schemes. We rephrase the three-party OC scheme as follows.

Definition 4 *An oblivious cast scheme consists of three parties: a sender S who sends a message m and two receivers A and B, such that the following conditions are satisfied:*

1. *The message m is received by exactly one of A and B, each with probability* $\frac{1}{2}$.
2. *The sender S cannot learn who got m.*
3. *The receiver who did not get m learns no information about m.*

First, we show how to construct OC by a COC-III scheme. We take the EQ-COC-III scheme as the example, where S owns a message m:

1. A and B randomly choose $x, y \in \{0, 1\}$ as their secrets, respectively.
2. All parties perform the EQ-COC-III scheme.
3. A will get m if $x = y$, or B will get m otherwise.

Since the probability that the two bits are equal is one half, each of A and B gets m with probability $\frac{1}{2}$. Furthermore, the sender S cannot learn who got m because S cannot learn the condition in the EQ-COC-III scheme. Also, because of the security of EQ-COC-III, the party who did not get m learns nothing about m.

Then we provide another simplified solution based on the homomorphic encryption scheme directly. The scheme is described in Figure 4.7. Assume that A, B and S are semi-honest and non-collusive parties, and A, B have their own key pairs (pk_A, sk_A), (pk_B, sk_B), respectively. The main idea is like the solution shown above. A and B choose random bits x and y, respectively, and send their encrypted random bits to S. After receiving the encrypted bits, S computes $v = m + r(x - y)$ and $v' = m + r'(x - \bar{y})$ in the encrypted form, and privately sends them to A and B, respectively, where r and r' are randomly chosen from the message space. Finally, A gets m if $x = y$ and B gets m otherwise.

We can see that the probability that the two bits are equal is one half. So each of A and B gets m with probability $\frac{1}{2}$. The sender S cannot learn who got m because of the semantic security of $\mathcal{E}$. If d or d' is not equal to 0, m will be masked by the randomly chosen value r or r'. Each receiver either gets m or learns nothing about m. The three conditions defined above are satisfied.

Since our constructions need only two rounds, they are more efficient in communication complexity than Blaze's scheme. Moreover, our schemes don't need any additional assumption on the communication channel.

4.5.2 Other Predicates

In addition to the basic predicates, we can design COC schemes for many other interesting predicates. For these predicates, the sender may need to perform multiplication on two messages encrypted by an additively homomorphic encryption scheme. However, there is no known encryption scheme with both additive and multiplicative homomorphism properties. So we use the BGN encryption scheme which can perform multiplication on two ciphertexts "one-time". In the setting of using threshold cryptosystems, the sender can even perform multiplication on two ciphertexts arbitrary times via some interactions [CDN01].

In fact, COC can be designed for a predicate of evaluating a bivariate polynomial $f(x, y)$. For example, to compute a public polynomial $f(x, y) = a_2x^2y^2 + a_1x^2y + a_0y$, the receivers send the encryptions of x, x^2, y and y^2

to the sender respectively. The sender then computes the polynomial by the following steps.

1. Perform the one-time multiplication on the encrypted messages such that $z_2 = x^2y^2$ and $z_1 = x^2y$.

2. Perform the constant multiplication: a_2z_2, a_1z_1 and a_0y.

3. Compute $f(x, y) = a_2z_2 + a_1z_1 + a_0y$.

After computing $f(x, y)$, the sender can embed messages by the result.

Alternatively, we can let one receiver hold the polynomial f and the other one hold the secret x. The sender sends messages by the result of $f(x)$. For example, for the "membership" predicate, one receiver encodes his set of secrets as a k-degree polynomial such that $f(x) = 0$ if and only if x belongs to the set, and the other receiver computes $x, x^2, \ldots, x^k$ for his secret x. The sender then sends the message to the receivers such that they get it if and only if $f(x) = 0$. This "membership" predicate can be used in our oblivious authenticated information retrieval application described in Section 1.

- Message sender S has a message $m \in \mathcal{M}$ and receivers A and B have their secrets x and y, respectively, where $x, y \in \{0,1\}^n$.
- Each of A, B and S has a secret key share corresponding to the public key pk.
- All parties should verify received proofs and encryptions of constants. Once a verification fails, the party terminates the protocol.

1. S posts $E_m = E_{pk}(m)$ and the corresponding NI-PPK proof.
2. A and B post $E_{x_{[i]}} = E_{pk}(x_{[i]}), E_{y_{[i]}} = E_{pk}(y_{[i]})$ and the corresponding NI-PPK and NI-PEB proofs, for $1 \leq i \leq n$.
3. S computes $E_0 = E_{pk}(0), E_1 = E_{pk}(1)$ and posts $(E_0, \gamma_0), (E_1, \gamma_1)$, where γ_0, γ_1 are the random factors used in E_0, E_1, respectively.
4. For each $i \in \{1, 2, \ldots, n\}$, all parties perform the following steps on the board:
   (a) Compute $E_{d_i} = E_{x_{[i]}} \boxplus (-1 \boxdot E_{y_{[i]}}), E_{d'_i} = E_{x_{[i]}} \boxplus E_{y_{[i]}} \boxplus (-1 \boxdot E_1)$.
   (b) Compute $E_{e_i} = E_{e_{i+1}} \boxplus E_{e_{i+1}} \boxplus E_{d_i}$, where $E_{e_{n+1}} = E_0$.
   (c) Compute $E_{f'_i} = E_{e_i} \boxplus (-1 \boxdot E_{d_i}) \boxplus E_{d'_i}$.
   (d) Each party privately computes and posts $E_{f_{Ai}} = r_{Ai} \boxdot E_{f'_i}, E_{f_{Bi}} = r_{Bi} \boxdot E_{f'_i}$ and $E_{f_{Si}} = r_{Si} \boxdot E_{f'_i}$, where $r_{Ai}, r_{Bi}, r_{Si} \in \mathbb{Z}_N$ are randomly chosen, respectively. Also, they post $E_{r_{Ai}}, E_{r_{Bi}}, E_{r_{Si}}$, the encryption of r_{Ai}, r_{Bi}, r_{Si}, and the corresponding NI-PCM proofs, respectively.
   (e) Compute $E_{f_i} = E_{f_{Ai}} \boxplus E_{f_{Bi}} \boxplus E_{f_{Si}}$.
   (f) All parties compute $E_{v_i} = E_m \boxplus E_{f_i}$.
5. All parties perform $(E_{v'_1}, E_{v'_2}, \ldots, E_{v'_n}) = \text{Mix-Net}(E_{v_1}, E_{v_2}, \ldots, E_{v_n})$.
6. S sends the partial decryptions of $(E_{v'_1}, E_{v'_2}, \ldots, E_{v'_n})$ to A and B.
7. A and B jointly decrypt the ciphertexts with S's partial decryptions and identify the correct message if existent.

Figure 4.6: MAL-INE-COC-I scheme against malicious parties

- Message sender S has a secret message $m \in \mathcal{M}$.
- Receivers A and B have their own key pairs $(pk_A, sk_A), (pk_B, sk_B)$ of $\mathcal{E}' = (G', E', D')$ respectively, and a common key pair (pk_R, sk_R) of $\mathcal{E} = (G, E, D)$.

1. A and B send $E_{pk_R}(x), E_{pk_R}(y)$ to S, respectively, where $x, y \in_R \{0, 1\}$.
2. S computes the following values via the homomorphic encryption scheme:
   (a) $d = x - y$, $d' = x + y - 1$.
   (b) $v = m + rd, v' = m + r'd'$, where $r, r' \in_R \mathcal{M}$.
3. S sends $E'_{pk_A}(E_{pk_R}(v))$ and $E'_{pk_B}(E_{pk_R}(v'))$ to A and B, respectively.
4. A and B decrypt the received messages and get the message m or a random value.

Figure 4.7: The Oblivious Cast Scheme

Chapter 5

Conclusion

In this thesis we discussed privacy-preserving data retrieval via introducing k-out-of-n oblivious transfer and conditional oblivious cast. We presented four very efficient OT_n^k schemes with unconditional security of either receiver or sender. The first two OT_n^k schemes with unconditional receiver's security are secure against semi-honest receivers in the standard model and malicious receivers in the random oracle model, respectively. The other two schemes with unconditional sender's security can be either generically constructed or efficiently performed. We also proposed an efficient Adpt-OT_n^k for adaptive queries. The essential technique is to reverse the order of key commitment and message commitment. In most previous schemes (including Semi-OT_n^k-I), the message commitments are dependent on the key commitments. Nevertheless, in our scheme Mal-OT_n^k-I, the message commitments are independent of the key commitments. Thus, the message commitments can be sent to R first.

Then we introduce a new notion of *conditional oblivious cast*, which ex-

tends conditional oblivious transfer to the three-party case. The definitions of this notion are given. We also provide COC implementations for some fundamental predicates, such as "equality", "inequality", and "greater than" predicates. By our schemes, we construct a new oblivious cast scheme without any additional assumption. We believe that COC is a fundamental primitive for secure multi-party computation.

Bibliography

[ADR02] Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *Proceedings of Advances in Cryptology - EUROCRYPT '01*, volume 2045 of *LNCS*, pages 119–135. Springer, 2001.

[AJL04] Andris Ambainis, Markus Jakobsson, and Helger Lipmaa. Cryptographic randomized response techniques. In *Proceedings of the Public Key Cryptography (PKC '04)*, volume 2947 of *LNCS*, pages 425–438. Springer, 2004.

[AR99] Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In *Proceedings of Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 65–79. Springer, 1999.

[BC97] Gilles Brassard and Claude Crépeau. Oblivious transfers and privacy amplification. In *Proceedings of Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *LNCS*, pages 334–347. Springer, 1997.

[BCR86a] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Proceedings of Advances in Cryptology - CRYPTO '86*, volume 263 of *LNCS*, pages 234–238. Springer, 1986.

[BCR86b] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. Information theoretic reductions among disclosure problems. In *Proceedings of 27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pages 427–437. IEEE, 1986.

[BCS96] Gilles Brassard, Claude Crépeau, and Miklós Sántha. Oblivious transfers and intersecting codes. *IEEE Transactions on Information Theory*, 42(6):1769–1780, 1996.

[BDSS02] Carlo Blundo, Paolo D'Arco, Alfredo De Santis, and Douglas R. Stinson. New results on unconditionally secure distributed oblivious transfer. In *Proceedings of Selected Areas in Cryptography - SAC '02*, volume 2595 of *LNCS*, pages 291–309. Springer, 2002.

[Bea92] Donald Beaver. How to break a "secure" oblivious transfer protocol. In *Proceedings of Advances in Cryptology - EUROCRYPT '92*, volume 658 of *LNCS*, pages 285–296. Springer, 1992.

[Bea95] Donald Beaver. Precomputing oblivious transfer. In *Proceedings of Advances in Cryptology - CRYPTO '95*, volume 963 of *LNCS*, pages 97–109. Springer, 1995.

[Bea96] Donald Beaver. Equivocable oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 119–130. Springer, 1996.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Proceedings of the 2nd Theory of Cryptography Conference (TCC 2005)*, volume 3378 of *LNCS*, pages 325–341. Springer, 2005.

[BK04] Ian F. Blake and Vladimir Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In *Proceedings of Advances in Cryptology - ASIACRYPT '04*, volume 3329 of *LNCS*, pages 515–529. Springer, 2004.

[Bla96] Matt Blaze. Oblevious key escrow. In *Proceedings of Information Hiding*, volume 1174 of *LNCS*, pages 335–343. Springer, 1996.

[BM89] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In *Proceedings of Advances in Cryptology*

- *CRYPTO '89*, volume 435 of *LNCS*, pages 547–557. Springer, 1989.

[BNPS01] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. Power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In *Proceedings of Financial Cryptography (FC '01)*, volume 2339 of *LNCS*, pages 319–338. Springer, 2001.

[Boe90] Bert den Boer. Oblivous transfer protecting secrecy. In *Proceedings of Advances in Cryptology - EUROCRYPT '90*, volume 473 of *LNCS*, pages 31–45. Springer, 1990.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Proceedings of the Public-Key Cryptography (PKC '03)*, pages 31–46. Springer, 2003.

[BPT84] Richard Berger, René Peralta, and Tom Tedrick. A provably secure oblivious transfer protocol. In *Proceedings of Advances in Cryptology - EUROCRYPT '84*, volume 209 of *LNCS*, pages 379–386. Springer, 1984.

[BS01] Olivier Baudron and Jacques Stern. Non-interactive private auctions. In *Proceedings of Financial Cryptography (FC '01)*, volume 2339 of *LNCS*, pages 364–378. Springer, 2001.

[Cac98] Christian Cachin. On the foundations of oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *LNCS*, pages 361–374. Springer, 1998.

[CC00] Christian Cachin and Jan Camenisch. Optimistic fair secure computation. In *Proceedings of Advances in Cryptology - CRYPTO '00*, volume 1880 of *LNCS*, pages 93–111. Springer, 2000.

[CCM98] Christian Cachin, Claude Crepeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *Proceedings of 39th Annual Symposium on Foundations of Computer Science (FOCS '98)*, pages 493–502. IEEE, 1998.

[CD97] Ronald Cramer and Ivan Damgård. Linear zero-knowledge - a note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 436–445. ACM, 1997.

[CDN01] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In *Proceedings of Advances in Cryptology - EUROCRYPT '01*, volume 2045 of *LNCS*, pages 280–299. Springer, 2001.

[Cha04] Yan-Cheng Chang. Single database private information retrieval with logarithmic communication. In *Proceedings of the 9th*

Australasian Conference on Information Security and Privacy (ACISP '04), volume 3108 of *LNCS*, pages 50–61. Springer, 2004.

[CK88] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions. In *Proceedings of 29th Annual Symposium on Foundations of Computer Science (FOCS '88)*, pages 42–52. IEEE, 1988.

[CM97] Christian Cachin and Ueli Maurer. Unconditional security against memory-bounded adversaries. In *Proceedings of Advances in Cryptology - CRYPTO '97*, volume 1294 of *LNCS*, pages 292–306. Springer, 1997.

[CMO00] Giovanni Di Crescenzo, Tal Malkin, and Rafail Ostrovsky. Single database private information retrieval implies oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '00*, volume 1807 of *LNCS*, pages 122–138. Springer, 2000.

[CNs07] Jan Camenisch, Gregory Neven, and abhi shelat. Simulatable adaptive oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '07*, volume 4515 of *LNCS*, pages 573–590. Springer, 2007.

[COR99] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional oblivious transfer and timed-release

encryption. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 74–89. Springer, 1999.

[Cré87] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In *Proceedings of Advances in Cryptology - CRYPTO '87*, volume 293 of *LNCS*, pages 350–354. Springer, 1987.

[Cré89] Claude Crépeau. Verifiable disclosure of secrets and applications. In *Proceedings of Advances in Cryptology - EUROCRYPT '89*, volume 434 of *LNCS*, pages 181–191. Springer, 1989.

[Cré97] Claude Crépeau. Efficient cryptographic protocols based on noisy channels. In *Proceedings of Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *LNCS*, pages 306–317. Springer, 1997.

[CS91] Claude Crépeau and Miklós Sántha. On the reversibility of oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '91*, volume 547 of *LNCS*, pages 106–113. Springer, 1991.

[CS06] Claude Crépeau and George Savvides. Optimal reductions between oblivious transfers using interactive hashing. In *Proceedings of Advances in Cryptology - EUROCRYPT '06*, volume 4004 of *LNCS*, pages 201–221. Springer, 2006.

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *Proceedings of the Public Key Cryptography (PKC '05)*, volume 3386 of *LNCS*, pages 172–183. Springer, 2005.

[CT06] Cheng-Kang Chu and Wen-Guey Tzeng. Conditional oblivious cast. In *Proceedings of the Public Key Cryptography (PKC '06)*, volume 3958 of *LNCS*, pages 443–457. Springer, 2006.

[CT08] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes. *Journal of Universal Computer Science*, 14(3):397–415, 2008.

[CvdGT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In *Proceedings of Advances in Cryptology - CRYPTO '95*, volume 963 of *LNCS*, pages 110–123. Springer, 1995.

[DFMS04] Ivan Damgård, Serge Fehr, Kirill Morozov, and Louis Salvail. Unfair noisy channels and oblivious transfer. In *Proceedings of Theory of Cryptography Conference (TCC '04)*, volume 2951 of *LNCS*, pages 355–373. Springer, 2004.

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model.

In *Proceedings of Theory of Cryptography Conference (TCC '04)*, volume 2951 of *LNCS*, pages 446–472. Springer, 2004.

[Din01] Yan Zong Ding. Oblivious transfer in the bounded storage model. In *Proceedings of Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 155–170. Springer, 2001.

[DKS99] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 56–73. Springer, 1999.

[DM99] Yevgeniy Dodis and Silvio Micali. Lower bounds for oblivious transfer reductions. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 42–55. Springer, 1999.

[DM02] Stefan Dziembowski and Ueli Maurer. Tight security proofs for the bounded-storage model. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing (STOC '02)*, pages 341–350. ACM, 2002.

[DR02] Yan Zong Ding and Michael O. Rabin. Hyper-encryption, and everlasting security. In *Proceedings of the 19th Annual Sympo-*

sium on Theoretical Aspects of Computer Science (STACS '02), volume 2285 of *LNCS*, pages 1–26. Springer, 2002.

[DS01] Paolo D'Arco and Douglas Stinson. Generalized zig-zag functions and oblivious transfer reductions. In *Proceedings of Selected Areas in Cryptography - SAC '01*, volume 2259 of *LNCS*, pages 87–102. Springer, 2001.

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In *Proceedings of Advances in Cryptology - CRYPTO '82*, pages 205–210. Plenum, 1982.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[FGMO01] Matthias Fitzi, Juan A. Garay, Ueli Maurer, and Rafail Ostrovsky. Minimal complete primitives for secure multi-party computation. In *Proceedings of Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 80–100. Springer, 2001.

[FMR96] Michael J. Fischer, Silvio Micali, and Charles Rackoff. A secure protocol for the oblivious transfer. *Journal of Cryptology*, 9(3):191–195, 1996.

[FPS00] Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. In *Proceedings of Financial Cryptography (FC '00)*, volume 1962 of *LNCS*, pages 90–104. Springer, 2000.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings of Advances in Cryptology - CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.

[GH07] Matthew Green and Susan Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *Proceedings of Advances on Cryptology - ASIACRYPT '07*, volume 4833 of *LNCS*, pages 265–282. Springer, 2007.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings of 41th Annual Symposium on Foundations of Computer Science (FOCS '00)*, pages 325–335. IEEE, 2000.

[GM97] Yael Gertner and Tal Malkin. Efficient distributed 1 out of n oblivious transfer. Technical Report MIT/LCS/TR-714, MIT Lab for Computer Science, April 1997.

[GM00] Juan Garay and Philip MacKenzie. Concurrent oblivious transfer. In *Proceedings of 41th Annual Symposium on Foundations of Computer Science (FOCS '00)*, pages 314–324. IEEE, 2000.

[GMY04] Juan Garay, Philip MacKenzie, and Ke Yang. Efficient and universally composable committed oblivious transfer and applications. In *Proceedings of Theory of Cryptography Conference (TCC '04)*, volume 2951 of *LNCS*, pages 297–316. Springer, 2004.

[GV87] Oded Goldreich and Ronen Vainish. How to solve any protocol problem - an efficiency improvement. In *Proceedings of Advances in Cryptology - CRYPTO '87*, volume 293 of *LNCS*, pages 73–86. Springer, 1987.

[Hai04] Iftach Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In *Proceedings of Theory of Cryptography Conference (TCC '04)*, volume 2951 of *LNCS*, pages 394–409. Springer, 2004.

[HCR02] Dowon Hong, Ku-Young Chang, and Heuisu Ryu. Efficient oblivious transfer in the bounded-storage model. In *Proceedings of Advances in Cryptology - ASIACRYPT '02*, volume 2501 of *LNCS*, pages 143–159. Springer, 2002.

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In *Proceedings of Advances in Cryptology - EUROCRYPT '05*, volume 3494 of *LNCS*, pages 96–113. Springer, 2005.

[IK97] Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *Proceedings of 5th Israel Symposium on Theory of Computing and Systems (ISTCS '97)*, pages 174–184. IEEE, 1997.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In *Proceedings of Advances in Cryptology - CRYPTO '03*, volume 2729 of *LNCS*, pages 145–161. Springer, 2003.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of 30th Annual Symposium on Foundations of Computer Science (FOCS '89)*, pages 230–235. IEEE, 1989.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21th Annual ACM Symposium on the Theory of Computing (STOC '89)*, pages 44–61. ACM, 1989.

[JS02] Ari Juels and Mike Szydlo. A two-server, sealed-bid auction protocol. In *Proceedings of Financial Cryptography (FC '02)*, volume 2357 of *LNCS*, pages 72–86. Springer, 2002.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In *Proceedings of Advances in Cryptology - EUROCRYPT '05*, volume 3494 of *LNCS*, pages 78–95. Springer, 2005.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC '88)*, pages 20–31. ACM, 1988.

[KK07] Kaoru Kurosawa and Takeshi Koshiba. Direct reduction of string (1,2)-ot to rabin's ot. Technical report, Cryptology ePrint Archive: Report 2007/046, 2007.

[KSV07] Mehmet S. Kiraz, Berry Schoenmakers, and José Villegas. Efficient committed oblivious transfer of bit strings. In *Proceedings of 10th Information Security Conference (ISC '07)*, volume 4779 of *LNCS*, pages 130–144. Springer, 2007.

[Lip03] Helger Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *Proceedings of Advances in Cryptology - ASIACRYPT '03*, volume 2894 of *LNCS*, pages 416–433. Springer, 2003.

[Lip05] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *Proceedings of 8th Information Security Conference (ISC '05)*, volume 3650 of *LNCS*, pages 314–328. Springer, 2005.

[Lip07] Helger Lipmaa. New communication-efficient oblivious transfer protocols based on pairings. Technical report, Cryptology ePrint Archive: Report 2007/133, 2007.

[LL06] Sven Laur and Helger Lipmaa. Consistent adaptive two-party computations. Technical report, Cryptology ePrint Archive: Report 2006/088, 2006.

[LL07] Sven Laur and Helger Lipmaa. A new protocol for conditional disclosure of secrets and its applications. In *Proceedings of Applied Cryptography and Network Security 2007 (ACNS '07)*, volume 4521 of *LNCS*, pages 207–225. Springer, 2007.

[LT05] Hsiao-Ying Lin and Wen-Guey Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In *Proceedings of Applied Cryptography and Network Security 2005 (ACNS '05)*, volume 3531 of *LNCS*, pages 456–466. Springer, 2005.

[Lu02] Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In *Proceedings of Advances*

in Cryptology - CRYPTO '02, volume 2442 of *LNCS*, pages 257–271. Springer, 2002.

[Mau90] Ueli Maurer. A provably-secure strongly-randomized cipher. In *Proceedings of Advances in Cryptology - EUROCRYPT '90*, volume 473 of *LNCS*, pages 361–373. Springer, 1990.

[MZV02] Yi Mu, Junqi Zhang, and Vijay Varadharajan. m out of n oblivious transfer. In *Proceedings of the 7th Australasian Conference on Information Security and Privacy (ACISP '02)*, volume 2384 of *LNCS*, pages 395–405. Springer, 2002.

[Nie07] Jesper Buus Nielsen. Extending oblivious transfers efficiently - how to get robustness almost for free. Technical report, Cryptology ePrint Archive: Report 2007/215, 2007.

[NNPV02] Ventzislav Nikov, Svetla Nikova, Bart Preneel, and Joos Vandewalle. On unconditionally secure distributed oblivious transfer. In *Proceedings of Progress in Cryptology - INDOCRYPT '02*, volume 2551 of *LNCS*, pages 395–408. Springer, 2002.

[NP99a] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC '99)*, pages 245–254. ACM, 1999.

[NP99b] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *Proceedings of Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 573–590. Springer, 1999.

[NP00] Moni Naor and Benny Pinkas. Distributed oblivious transfer. In *Proceedings of Advances in Cryptology - ASIACRYPT '00*, volume 1976 of *LNCS*, pages 200–219. Springer, 2000.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms (SODA '01)*, pages 448–457. ACM/SIAM, 2001.

[NPS99] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of the 1st ACM conference on Electronic commerce (EC '99)*, pages 129–139. ACM, 1999.

[NR94] Valtteri Niemi and Ari Renvall. Cryptographic protocols and voting. In *Results and Trends in Theoretical Computer Science*, volume 812 of *LNCS*, pages 307–317. Springer, 1994.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, pages 458–467. IEEE, 1997.

[NSNK04] Lan Nguyen, Reihaneh Safavi-Naini, and Kaoru Kurosawa. Verifiable shuffles: A formal model and a paillier-based efficient construction with provable security. In *Proceedings of Applied Cryptography and Network Security 2004 (ACNS '04)*, volume 3089 of *LNCS*, pages 61–75. Springer, 2004.

[OK04] Wakaha Ogata and Kaoru Kurosawa. Oblivious keyword search. *Journal of Complexity*, 20(2-3):356–371, 2004.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[SCP95] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Zero-knowledge arguments and public-key cryptography. *Information and Computation*, 121(1):23–40, 1995.

[SP90] Alfredo De Santis and Giuseppe Persiano. Public-randomness in public-key cryptography. In *Proceedings of Advances in Cryptology - EUROCRYPT '90*, volume 473 of *LNCS*, pages 46–62. Springer, 1990.

[SS90] Arto Salomaa and Lila Santean. Secret selling of secrets with several buyers. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 42:178–186, 1990.

[Sta96] Markus Stadler. Publicly verifiable secret sharing. In *Proceedings of Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 190–199. Springer, 1996.

[Ste98] Julien P. Stern. A new and efficient all or nothing disclosure of secrets protocol. In *Proceedings of Advances in Cryptology - ASIACRYPT '98*, volume 1514 of *LNCS*, pages 357–371. Springer, 1998.

[Tze04] Wen-Guey Tzeng. Efficient 1-out-of-n oblivious transfer schemes with universally reusable parameters. *IEEE Transactions on Computers*, 53(2):232–240, 2004.

[Vad03] Salil P. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In *Proceedings of Advances in Cryptology - CRYPTO '03*, volume 2729 of *LNCS*, pages 61–77. Springer, 2003.

[Wie83] Stephen Wiesner. Conjugate coding. *ACM SIGACT News*, 15(1):78–88, 1983.

[Wul07] Jürg Wullschleger. Oblivious-transfer amplification. In *Proceedings of Advances in Cryptology - EUROCRYPT '07*, volume 4515 of *LNCS*, pages 555–572. Springer, 2007.

[WW06] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In *Proceedings of Advances in Cryptology - EUROCRYPT '06*, volume 4004 of *LNCS*, pages 222–232. Springer, 2006.

[WZW03] Qian-Hong Wu, Jian-Hong Zhang, and Yu-Min Wang. Practical t-out-n oblivious transfer and its applications. In *Proceedings of 5th International Conference on Information and Communications Security (ICICS'03)*, volume 2836 of *LNCS*, pages 226–237. Springer, 2003.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of 27th Annual Symposium on Foundations of Computer Science (FOCS '86)*, pages 162–167. IEEE, 1986.